

KARELIA-AMMATTIKORKEAKOULU

Tieto- ja viestintätekniikan koulutusohjelma

Jukka Pirinen

LISÄTYN TODELLISUUDEN TEKNOLOGIOIDEN  
HYÖDYNTÄMINEN MYYNNISSÄ

Opinnäytetyö

Huhtikuu 2018



**OPINNÄYTETYÖ**  
**Huhtikuu 2018**  
**Tieto- ja viestintätekniikan**  
**koulutusohjelma**  
Tikkarinne 9  
80200 JOENSUU  
+358 13 260 600 (vaihde)

Tekijä

Jukka Pirinen

Nimeke

Lisätyn todellisuuden teknologioiden hyödyntäminen myynnissä

Toimeksiantaja

Tulikivi Oyj

Tiivistelmä

Opinnäytetyön tavoitteena oli lisätyn todellisuuden ratkaisun kehittäminen tuotemalliston markkinointia ajatellen, sisältäen tekniset välinevalinnat ja toimintaprosessin määrittelyn. Aihe valikoitui toimeksiantajan mielenkiinnosta käyttää lisättyä todellisuutta myynninedistämistarkoituksiin, myös opinnäytetyöntekijän aiempien projektien pohjalta syntynyt henkilökohtainen kiinnostus aihepiiriä kohtaan vaikutti työn muotoutumiseen.

Työssä käydään läpi lisätyn todellisuuden mahdollisuuksia sekä tulevaisuuden visioita, pääasiassa mobiilisovellusten ja markkinoinnin näkökulmasta. Lisätty todellisuus on aiheena ajankohtainen, etenkin sovellusten ja käytettävien laitteistojen kehittyessä teknisesti paremmiksi. Tulevaisuudessa suosio aihepiiriä kohtaan kasvaa. Työssä käydään myös läpi lisätyn todellisuuden mahdollisuuksia markkinoinnin näkökulmasta, sekä tutustutaan Unity-pelimoottoriin ja sen avulla toteutettujen lisätyn todellisuuden sovelluksien kehitykseen.

Lopputuloksena opinnäytetyöstä valmistui katsaus lisätyn todellisuuden sovellusten tulevaisuudesta, sekä käytöstä mainonnan apuvälineenä. Tuloksena oli myös mobiilisovellus Android-käyttöjärjestelmälle, jolla voidaan esitellä toimeksiantajan tuotantokäyttöön tarkoitettua mallia, josta toteutettiin lisätyn todellisuuden esittelymalli.

Kieli

suomi

Sivuja 59

Liitteet 5

Liitesivumäärä 12

Asiasanat

lisätty todellisuus, mobiilisovellukset, myynti, markkinointi



**THESIS**  
**April 2018**  
**Degree Programme in Information and**  
**Communications Technology**  
Tikkarinne 9  
80200 JOENSUU  
FINLAND  
+358 13 260 600 (switchboard)

Author

Jukka Pirinen

Title

Using Augmented Reality Technologies for Sales Promotion

Commissioned by

Tulikivi Oyj

Abstract

The purpose of this thesis was to develop an augmented reality solution for the marketing of the product models, including technical tools selection and definition of the operating process. The subject was selected by the client's interest in using augmented reality for sales promotional purposes. Furthermore, the author's personal interest in this subject based on the previous projects contributed to the formation of this work.

The thesis examines the possibilities of augmented reality and its visions of the future, mainly from the perspective of mobile applications and marketing. Augmented reality as the subject is topical, especially as the applications and hardware are becoming technically more advanced. In the future, the popularity of the topic will increase. The thesis also investigates the possibilities of augmented reality from the marketing perspective, as well as introduces Unity game engine and the development of augmented reality applications with it.

As a result of this thesis, a review was completed of the future of augmented reality applications and their use as a tool for advertising. The result also included a mobile application for the Android operating system, which can present the client's product by using the means of augmented reality.

Language

Finnish

Pages 59

Appendices 5

Pages of Appendices 12

Keywords

augmented reality, mobile applications, sales, promotional

# Sisältö

Lyhenteet ja termit .....	6
1 Johdanto .....	8
2 Lisätty todellisuus.....	9
2.1 Lisätyn todellisuuden historia.....	11
2.2 Lisätty todellisuus nykypäivänä.....	12
2.3 Lisätyn todellisuuden tulevaisuus .....	12
2.4 Lisätty todellisuus markkinoinnissa.....	13
2.5 Konenäkö ja seuranta.....	17
2.6 Laitteet.....	18
2.6.1 Päälaitteet.....	18
2.6.2 Mobiililaitteet.....	19
2.6.3 Projisoivat laitteet.....	20
3 Toimeksianto .....	21
3.1 Toimeksiantaja.....	21
3.2 Tavoitteet.....	22
3.3 Lähtökohdat.....	22
4 Toteutustavan valinta.....	23
4.1 Unity-pelimoottori.....	24
4.1.1 Unity-lisenssi.....	25
4.1.2 Unity-käsitteitä .....	26
4.1.3 3D-formaatit .....	29
4.2 Google ARCore .....	30
4.3 Apple ARKit .....	32
4.4 Vuforia AR SDK.....	32
4.5 SolidWorks.....	34
4.6 3D-mallinnusohjelmat .....	35
5 Sovelluksen toteutus.....	37
5.1 Käytetyt sovellukset ja laitteet.....	37
5.2 Mallin siirto.....	38
5.3 Vuforia .....	39
5.3.1 Vuforia Developer .....	39
5.3.2 Vuforia Unity-pelimoottorissa.....	40
5.4 Malli Unity-pelimoottorissa .....	42
5.4.1 Materiaalit .....	42
5.4.2 Luukun toiminnallisuus .....	44
5.4.3 Tuli.....	44
5.5 Kohtaukset.....	45
5.5.1 Aloitusvalikko .....	46
5.5.2 Sovellus-kohtaus .....	47
5.6 Valo ja varjostukset.....	48
5.7 Testaus.....	49
5.8 Sovelluksen käännös.....	50
6 Tulokset .....	50
6.1 Prosessi.....	52
6.2 Yhteenveto.....	53
7 Pohdinta.....	54
7.1 Toteutuksen tarkastelu .....	54

7.2	Tuotteen hyödyntäminen .....	55
7.3	Jatkotutkimus ja kehitysideat .....	55
Lähteet	.....	57

#### Liitteet

Liite 1	Asennusohje
Liite 2	OpenDoor.cs
Liite 3	MenuScript.cs
Liite 4	SmoothCamera.cs
Liite 5	Tulikivi_tunnus_BLACK.PNG

## Lyhenteet ja termit

3D	Three dimensional, 3D-grafiikka, kolmiulotteinen esitystapa geometrisestä datasta.
APK	Android application package, Android-sovellusten tiedostotyyppi.
AR	Augmented Reality, lisätty todellisuus.
ARCore	Googlen Android-käyttöjärjestelmille kehittämä lisätyn todellisuuden kehitysalusta.
ARKit	Applen iOS-käyttöjärjestelmille kehittämä lisätyn todellisuuden kehitysalusta.
Blender	3D-grafiikan ilmainen ja alustariippumaton mallinnusohjelma.
CAD	Computer-aided Design, tietokoneavusteinen suunnittelu.
Frame	Tietyllä hetkellä renderoitu kuva.
GPS	Global Positioning System, satelliittipaikannusjärjestelmä.
MR	Merged Reality, Mixed Reality, yhdistetty todellisuus.
Pelimoottori	Pelimoottori on videopelien ohjelmoinnin apuväline. Pelimoottorin avulla pelien ja sovellusten kehittäminen on nopeampaa ja helpompaa.
Pikseli	Kuvapiste, kuvan pienin yksittäinen osa.
PTC	Ohjelmistoyritys, aiemmalta nimeltään Parametric Technology Corporation.
Renderointi	Kuvan ohjelmallinen luonti mallista, usein kolmiulotteinen objekti.
Resoluutio	Kuvan muodostavien pikseleiden määrä. Pikseleiden määrä vaakasuunnassa kerrottuna pikseleiden määrällä pystysuunnassa.
Scene	Sovelluksen kohtaus, näkymä Unity-pelimoottorissa.
Script	Skripti, komentosarja, lyhyt tietokoneen ohjelma. Unity-objektien toiminnallisuus toteutetaan skripteillä.
SDK	Software development kit, sovelluskehityspaketti.

SLAM	Simultaneous localization and mapping, yhtäaikainen paikannus ja kartoitus.
Solidworks	Dassault Systèmes yhtiön CAD-Suunnitteluohjelmisto [1].
Unity	Alustariippumaton pelimoottori.
VR	Virtual Reality, virtuaalinen todellisuus. Tietokonesimulaation tuottamien aistimusten avulla luotu keinotekoinen ympäristö [2].
Vuforia	Lisätyn todellisuuden sovelluskehitysalusta, PTC:n kehittämä.

## 1 Johdanto

Opinnäytetyön aiheena oli lisätyn todellisuuden ratkaisun kehittäminen, jota voitaisiin hyödyntää toimeksiantajan tuotemalliston markkinoinnissa myynnin edistämiseksi. Aihe muotoutui toimeksiantajan kiinnostuksesta lisätyn todellisuuden esittelysovellusta kohtaan, jolla voitaisiin esitellä konkreettisesti suuria tuotteita vaivattomammin. Myös opinnäytetyöntekijän aiempien projektien pohjalta syntynyt henkilökohtainen kiinnostus aihepiiriä kohtaan vaikutti aiheen ja projektin lopulliseen muotoutumiseen.

Opinnäytetyössä käydään läpi yleisellä tasolla käsitteitä lisätystä todellisuudesta ja sen erilaisten sovellutusten mahdollisuuksista, mukaan lukien käyttö markkinoinnin näkökulmasta, sekä aihepiirin tulevaisuuden visioita. Dokumentoinnin kohteena on myös varsinaisen lisätyn todellisuuden sovelluksen kehitys- ja toteutusvaiheita, aina toimeksiantajan suunniteltupöydältä tulevasta rakennepiirustuksesta, prototyyppiseen esittelysovellukseen asti.

Lisätyn todellisuuden sovelluksien ja käytettävien laitteistojen kehittyessä koko ajan teknisesti paremmiksi, kyseiset sovellukset ovat nostamassa suosiotaan tulevaisuutta ajatellen, niin sovelluskehittäjien, kuin sovellusten loppukäyttäjien kannalta. Työn toteutuksen aikana moni työssä käytetty, sekä tutkinnan kohteena ollut ohjelmistokehityspaketti sai uusia päivityksiä ja toiminnallisuuksia, joten kehityksen aihepiirin parissa voitiin todeta konkreettisesti ottavan tällä hetkellä todella isoja harppauksia eteenpäin, mutta vielä ei ole liian myöhäistä hypätä mukaan lisätyn todellisuuden junaan.



## 2 Lisätty todellisuus

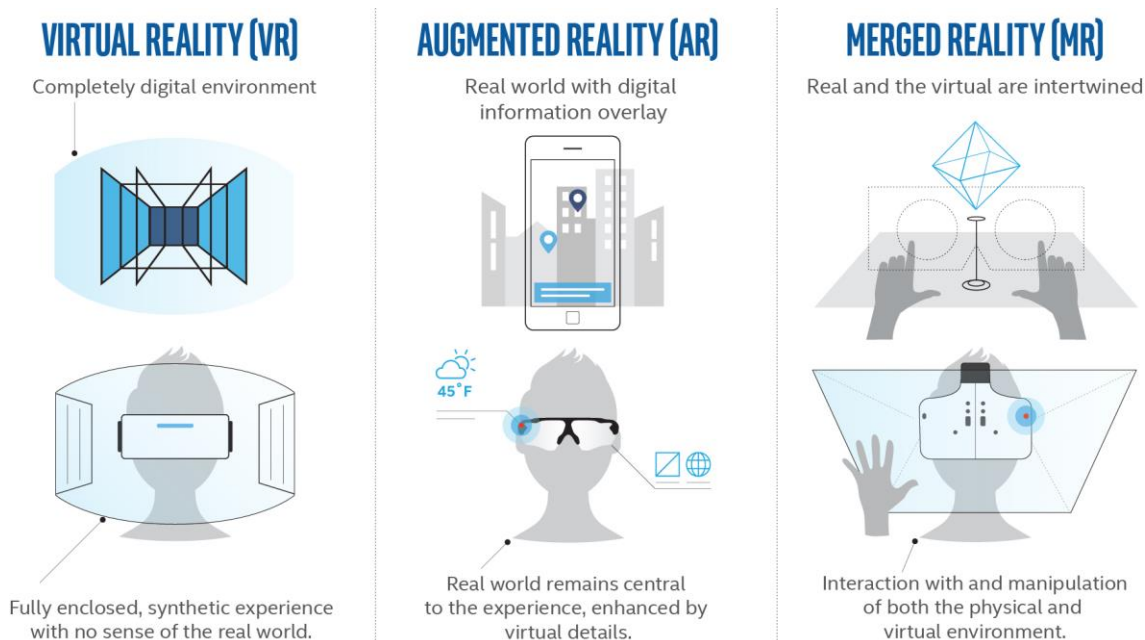
Laajennettu todellisuus on laajin ja yksinkertaisin nimitys joukolle teknologioita, joilla yhdistetään tietokoneen avulla luotua materiaalia, joko todellisiin ympäristöihin eli lisättyyn todellisuuteen, myös AR (engl. augmented reality), tai virtuaalisiin ympäristöihin eli virtuaaliseen todellisuuteen, myös VR (engl. virtual reality). Mukaan voidaan lukea myös yhdistetty todellisuus eli MR (engl. mixed reality, merged reality), jossa voidaan vaikuttaa sekä todellisen, että virtuaalisen maailman elementteihin reaaliaikaisesti [3].

Lisätty todellisuus on avoin simulaatio, joka pitää sisällään digitaalista dataa ja oikean maailman elementtejä, joita yhdistetään reaaliajassa. Tämä liittyy yleensä 3D-sisällön lisäämiseen kameran välittämään videokuvaan, jonka avulla käyttäjälle luodaan mahdollisimman aito vaikutelma lisätyn elementin kuulumisesta reaali maailmaan. Reaaliaikaisuus onkin myös yksi tärkeimmistä kriteereistä lisätyn todellisuuden määrittelyssä, joskaan lisätyn todellisuuden ei välttämättä tarvitse olla aina visuaalista. Käyttäjiä voidaan opastaa myös esimerkiksi äänen avulla tai antamalla haptisia, tuntoaistiin perustuvia ärsykeitä.

Lisätyn todellisuuden kehitysympäristöjä on ollut jo 90-luvun loppupuolelta saakka, mutta varsinkin matkapuhelinten ja taulutietokoneiden nopea lisääntyminen on kiihdyttänyt myös lisätyn todellisuuden kehitystä sekä lisännyt kuluttajien kiinnostusta aiheita kohtaan. Joskaan lisätyn todellisuuden mahdollisuudet käsilaitteille eivät ole vielä saavuttaneet koko potentiaaliaan, Applen ja Googlen vasta hypätessä kehitykseen mukaan vuoden 2017 aikana. Applen julkaistua oman ARKit-kehitysalustansa kesäkuussa ja Googlen oman ARCore-alustan elokuussa 2017. [4, 8.]

Lisätty todellisuus on kuitenkin syytä erottaa virtuaalitodellisuudesta, jossa käyttäjä kokee ainoastaan keinotekoista syötettä, vailla reaali maailman vaikutusta koettavaan asiaan. Virtuaalitodellisuus tarvitsee myös käyttöönsä varten erikoislaitteita, esimerkiksi päässä pidettävää näyttöä, kun taas lisätyn todellisuuden käyttö on mahdollista jo vaikkapa matkapuhelimenkin avulla. Tästä syystä

virtuaalitodellisuus vaatiikin käyttäjältään enemmän sitoutumista, kun lisätyn todellisuuden avulla voidaan tuoda virtuaaliset asiat käyttäjän saataville helpommin (kuva 1).



Kuva 1. VR, AR, MR [5].

Lisättyä todellisuutta käytetään muun muassa liiketoiminnan markkinoinnissa, koulutuksessa, teollisuudessa, jälleenmyynnissä, viihteessä, sekä 3D- että CAD-suunnittelukäytössä. Lisätyn todellisuuden avulla voidaan jakaa informaatiota tehokkaammin, esimerkiksi näyttää toisella puolella maailmaa oleva tuote asiakkaalle ja luoda näin vaikutelma paikalla olost.

Tässä opinnäytetyössä käsiteltävä lisätty todellisuus keskittyy ainoastaan todelliseen ympäristöön lisättävien, reaaliaikaisesti luotujen virtuaalielementtien käsittelyyn tietokoneen avulla. Luodut elementit yhdistetään laitteen kameran välittämään kuvasyötteeseen, joten käyttäjä näkee todelliseen maailmaan sijoitettuja virtuaalisia elementtejä. Käyttäjälle on tarkoitus luoda vaikutelma todellisten ja tietokoneella luotujen objektien olevan yhtä todellisia.

” Laajennetun todellisuuden voima on, että se mahdollistaa koneiden toimivan kognitiivisella tasolla, jota teemme ihmisinä. Antaen meille ihmisinä luonnollisen vuorovaikutuksen teknologiamme kanssa.” [6]

## 2.1 Lisätyn todellisuuden historia

Lisätty todellisuus käsitteenä mainitaan ensimmäistä kertaa vuonna 1901 romaanissa *The Master Key* [7]. Teoksessa on maininta elektronisista silmälasista, joiden avulla lisätään tietoa käyttäjän näkökenttään.

”Annan sinulle henkilön merkkajaan. Se koostuu tästä silmälasien parista. Kun käytät niitä, jokainen jonka tapaat, on merkitty otsaan kirjaimella, osoittaen hänen luonteensa.” [7]

Vuonna 1962 valmistunut Sensorama [8] on laite, joka yhdistää kuvaa, ääntä, tärinää ja tuoksuja. Se mainitaankin usein lisättyä todellisuutta käsittelevissä tutkielmissa ensimmäisenä luotuna laitteena, sen esittäessä käyttäjälle kuitenkin ainoastaan laitteen avulla luotuja ärsykeitä. Laite voidaan lukea näin ollen kuuluvaksi enemmän virtuaalitodellisuuden kategoriaan, kuin varsinaiseksi lisätyn todellisuuden sovellutukseksi.

Vuonna 1968 Ivan Sutherland rakensi ensimmäisen päässä pidettävän näytön [9], jolla voitiin lisätä käyttäjän näkökenttään objekteja. Tätä voidaankin pitää ensimmäisenä varsinaisen lisätyn todellisuuden laitteena.

Lisättyä todellisuutta ja sen sovellutuksia on ollut olemassa jo pitkään. Mutta vasta viime vuosina tekniikan kehittyessä, se on alkanut nostamaan suosiotaan myös suuremman yleisön keskuudessa.

## 2.2 Lisätty todellisuus nykypäivänä

Nykyään lähes kaikilla on tehokas tietokone mukanaan, älypuhelin, joissa on tarkat kamerat, sekä tehokkaat prosessorit ja näin ollen mahdollisuus käsitellä yhä kehittyvämpiä ohjelmistoja [10, 23]. Suurin osa tällä hetkellä olemassa olevista sovelluksista onkin keskittynyt lähinnä viihdeohjelmistoihin ja peleihin, jotka ovatkin useasti kehityksen kärjessä. Mutta enenevässä määrin yritykset ovat ottaneet käyttöön myös teollisuus- ja markkinointikäyttöön soveltuvia lisätyn todellisuuden sovelluksia.

Tällä hetkellä suurimpina ja tunnetuimpina kehitysalustoina voidaan lisätyn todellisuuden sovellusten kehityksessä mainita ainakin PTC:n Vuforia, Applen ARKit, Googlen ARCore, sekä Microsoftin HoloLens. Näistä Vuforia-kehitysalustaa voitaisiin pitää käytännöllisimpänä, sen ollessa alustasitoutumaton sekä perusominaisuuksiltaan maksuton.

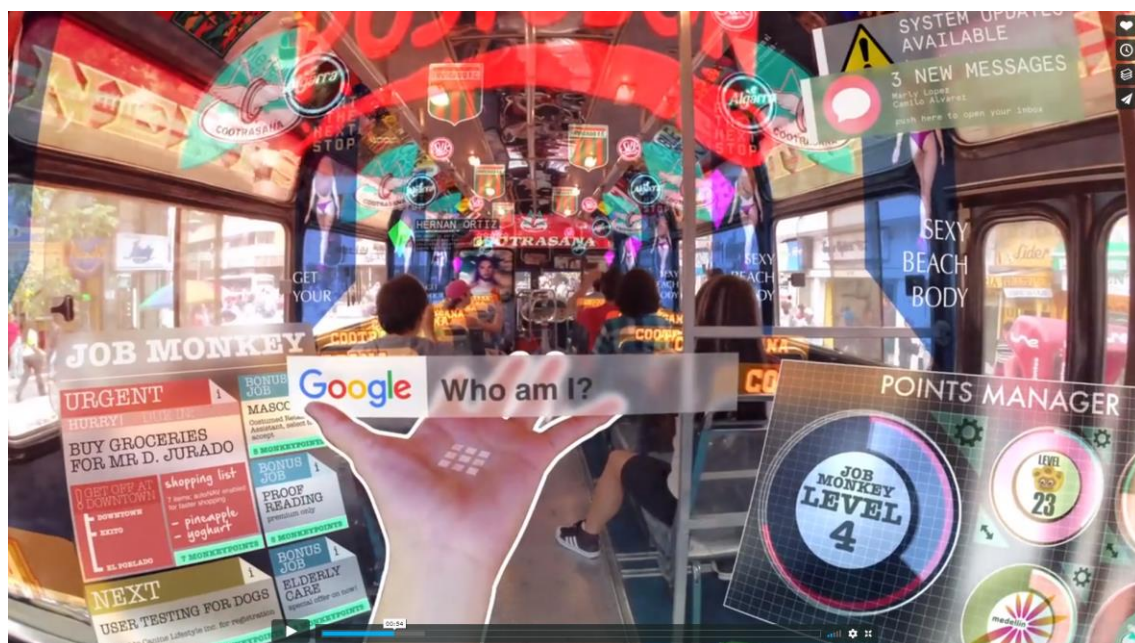
## 2.3 Lisätyn todellisuuden tulevaisuus

Lisätty todellisuus on vaativaa laitteistoa kohtaan, mutta teknologian yhä kehittyessä paremmaksi voidaan olettaa uusien sovellusten tulevan kasvavissa määrin markkinoille, myös viihdekäytön ulkopuolelle. Näin ollen myös sovellusten käyttäjämäärien ollessa nousussa suosio kasvanee jatkossa lisätyn todellisuuden saralla.

Lisätty todellisuus tulee luultavasti olemaan myös suositumpaa kuin esimerkiksi virtuaalinen todellisuus, joka vaatii teknisesti tehokkaampia laitteita sekä aina jonkinlaisen päälaitteen. Kun taas lisätyn todellisuuden teknologioissa useimmiten riittävä laite on käyttäjän oma älypuhelin tai taulutietokone ja näin ollen saavutetaan sovellusten käytön vaivattomuus.

Ohjaaja Keiichi Matsudan tekemä lyhytelokuva (kuva 2) voi olla eräänlainen tulevaisuuden visio lisätystä todellisuudesta. Videolla nähdään käyttäjän näkemää lisättyä todellisuutta, joka tulvii käyttäjän näkökenttään, kaupunkikuvan ollessa

täytetty mitä erilaisimmilla syötteillä. Etenkin käytettävien laitteiden ja näyttöjen pienentyessä silmälasien tai jopa piilolinssien kokoon, saadaan avautumaan kokonaan uusia sovelluskohteita arkipäiväisiinkin kohteisiin.



Kuva 2. Hyper-Reality [11].

## 2.4 Lisätty todellisuus markkinoinnissa

Useat yritykset käyttävät jo suhteellisen yksinkertaisia sovelluksia, joilla esimerkiksi voidaan kuvata yrityksen mainoskuvaa ja sovellus näyttää lisäinformaatiota aiheesta renderoimalla kuvan päälle lisätyn todellisuuden mallin. Sovelluksia voidaan käyttää monissa käyttötarkoituksissa, myös näyttelyissä tai myymälöissä: myynninedistämistarkoituksessa tai puhtaana viihteenä [10, 107].

”Puhelin tulee todennäköisesti olemaan valtavirran alusta, joissa lisätyn todellisuuden ominaisuudet tulevat käyttöön, sen sijaan että ihmiset pitäisivät laseja päässään”. [12, 16.]

## **Tuotetietojen laajennus**

Tuotteiden pakkauksia, yritysten logoja tai muita esittelymateriaaleja voidaan käyttää merkkeinä, joiden päälle tuodaan lisäinformaatiota lisätyn todellisuuden avulla. Myös esimerkiksi käyntikortin käyttämistä sovelluksen pohjakuvana saadaan aikaan näyttävyyttä ja huomioarvoa. Nykyään yrityksien käytössä näkyvät usein QR-koodit, jotka vievät esimerkiksi haluttuun nettiosoitteeseen, nämä voitaisiinkin korvata tulevaisuudessa merkkikuvilla, joiden päälle voidaan rakentaa lisättyä todellisuutta.

## **Tuotteiden esittely**

Yrityksillä jotka myyvät fyysisiä tuotteita, kuten huonekaluja, koristeita tai vaikka ruokaa, voidaan antaa ihmisille parempi käsitys siitä, miten tuote sopii todelliseen maailmaan. Käyttämällä lisättyä todellisuutta, asiakas saa myös paremman mielikuvan tuotteiden mittakaavasta ja niiden visuaalisesta ilmeestä. [13]

Tuotetta voidaan esitellä missä tahansa, sitä voidaan tarkastella eri suunnista, sekä tuotetta on helppo muokata ja vaihdella. Tuotepalautukset vähentyvät asiakkaan saadessa paremman kuvan tuotteesta jo ennen ostopäätöksen tekemistä.

## **Ongelmanratkaisun apuväline**

Esimerkkinä ongelmanratkaisun apuvälineestä voitaisiin mainita lisätyn todellisuuden sisustussovellus, jonka avulla asiakas tai suunnittelija näkee tuotteiden yhteensopivuuden todellisen maailman kohteiden tai toisten virtuaaliobjektien kanssa. Näin voidaan vähentää todellisessa maailmassa tapahtuvia testauksia, jotka vaativat aikaa ja resursseja.

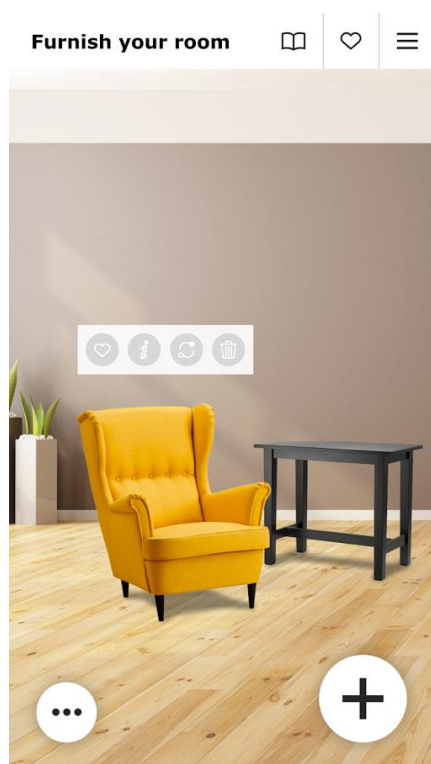
## **Valokuvat ja sosiaalinen media**

Sosiaalisen median palvelut; Facebook, Snapchat ja Instagram käyttävät jo lisätyn todellisuuden kuvaussovelluksia, joiden avulla käyttäjät voivat lisätä valo- ja videokuviinsa kuvan ottohetkellä uusia elementtejä. Tätä ominaisuutta käytetään

myös tuotemerkkien mainostamiseen, yritysten luodessa sovelluksiin omia suodattimiaan, joilla kuvauksen aikana voidaan lisätä mainostietoa lopputuloksena saatavaa kuvaan.

Joten markkinoijat kaikkialla: Ole valmis käyttämään lisättyä todellisuutta, muuttaaksesi ikuisesti sitä, kuinka asiakkaasi ovat vuorovaikutuksessa tuotemerkkisi kanssa. Eräänä päivänä puhelimemme kuitistuvat piilolinssien kokoon ja lisätyn todellisuuden kokemukset ovat hallitsevia jokaisessa hereillä olon hetkessä. Kun tämä tapahtuu, voit sanoa että olit siellä alusta asti, seuranasi ARKit, ARCore ja Pichachu. [14].

### IKEA-kuvastosovellus

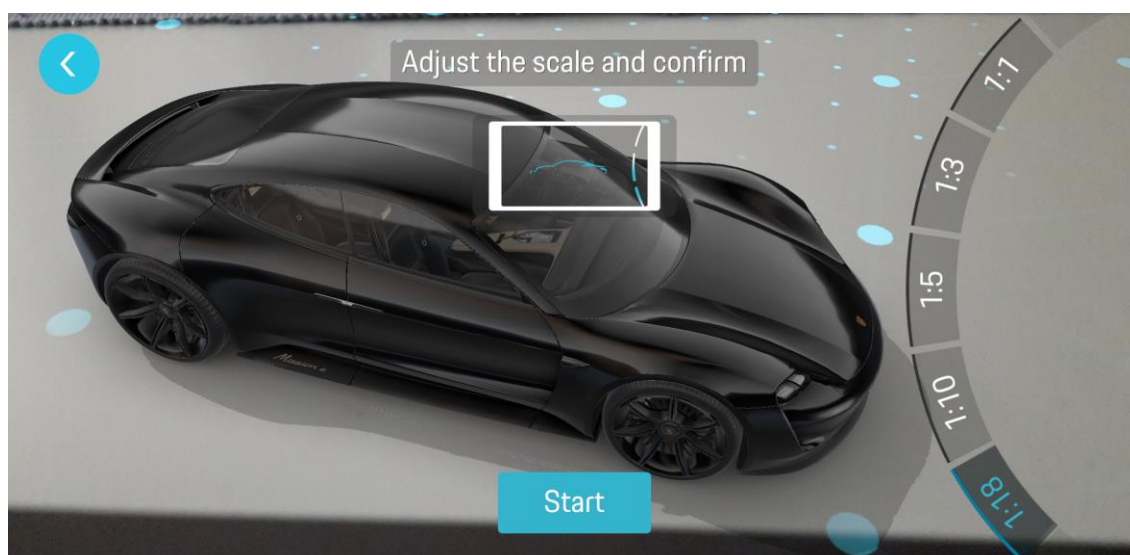


Kuva 3. IKEA-Kuvasto [15].

Esimerkkinä myynnin ja tuotteiden esittelyn kannalta voitaisiin mainita IKEA-huonekalumyymäläketjun kuvastosovellus (kuva 3). Sovelluksella on mahdollisuus tarkkailla mallistoa 3D-ominaisuuden avulla käyttäjän haluamissa tiloissa. Sovelluksen avulla mahdollinen asiakas saa paremman käsityksen miltä varsinainen tuote näyttäisi tilassa ja miten se sopisi muuhun sisustukseen. Sovelluksen iOS-käyttöjärjestelmälle toteutetussa versiossa käyttäjä voi skannata huoneen kokoa, jonka jälkeen tarkasteltavat mallit sopivat paremmin ympäristöönsä. Android-käyttöjärjestelmän toteutetussa versiossa ei huoneen kokoa voitu sovelluksen testaushetkellä havainnoida erikseen, joten toisinaan huonekalut jäivät hie- man irrallisiksi ympäristöstään.

## Porsche-esittelysovellus

Autonvalmistaja Porschen tarjoama Porsche Mission E AR [16] on lisätyn todellisuuden sovellus, jonka avulla esitellään valmistajan sähköautomallia. Sovellus on toteutettu käyttäen ARCore- ja ARKit-teknologioita, riippuen käyttöjärjestelmästä, jolla sovellusta käytetään. Sovelluksella käyttäjä voi nähdä mallin joko luonnollisessa koossa tai eri mittakaavassa olevina pienoismalleina (kuva 4). Myös mallin väriä voidaan muuttaa, sen sisällä olevaan teknologiaan voidaan tutustua tarkemmin, sekä sillä voidaan ajaa pieniä lenkkejä niin eteen- kuin taaksepäin.



Kuva 4. Porsche Mission E AR -sovellus (Kuva: Jukka Pirinen).

Lisätyn todellisuuden sovelluksia voidaan pitää erittäin hyvänä keinona tutustua tuotteeseen ja siihen miltä valmis tuote tulisi näyttämään asiakkaan käytössä. Mitä suurempi ja arvokkaampi tuote on, sitä enemmän asiakas joutuu harkitsemaan päätöstään. Ja näin ollen päätöksenteon avuksi voidaan tarjota lisäinformaatiota tuotteesta edellä mainittujen kaltaisten sovellusten avulla.



## 2.5 Konenäkö ja seuranta

Konenäkö on lyhykäisyydessään kameran välittämää kuvaa kohteesta, joka tulkitaan sovelluksen kuvankäsittelyominaisuudella. Tulkitun tiedon avulla toteutetaan haluttuja toimenpiteitä, esimerkiksi pintojen tai kuvioden tunnistamista.

Lisätyn todellisuuden sovelluskehityksessä ja tästä näkökulmasta tarkasteltuna konenäkö ja seuranta voidaan jakaa eri kategorioihin: merkkipohjaiseen, merkittömään ja paikannukseen perustuvaan kohteen tunnistukseen. Sovelluksen käyttökohteesta, käyttöympäristöstä ja käytettävästä laitteesta riippuen voidaan valita kulloiseenkin käyttöön parhaiten toimiva menetelmä tai näiden menetelmien yhdistelmä.

### SLAM-seuranta

SLAM-seuranta (engl. simultaneous localization and mapping) kartoittaa ja mallintaa reaaliaikaisesti ympäristöä, muuntaen visuaalisen datan tietokoneen ymmärtämiksi pisteiksi. Näitä pisteitä voidaan hyödyntää rakennusten sisätilanpaikannuksessa, sekä objektien lisäämiseksi ympäröivää tilaan. Slam-teknologiaa ja laitteen antureita hyödyntämällä voidaan myös sisätilojen paikannusta ja mallinnusta käyttää tehokkaammin hyödyksi.

Tätä menetelmää käyttävä sovellus pyrkii tunnistamaan ympäristössä näkyviä, helposti tunnistettavia piirteitä, kuten esineiden ja asioiden kulmia tai tasaisia pintoja. Varsinkin ympäröivän maailman valaistus, että sen aiheuttamat heijastukset ja varjot voivat aiheuttaa vaikeuksia tunnistukselle. Mutta laitteiden kehittyessä, uusimmat mobiililaitteetkin pystyvät jo käsittelemään sovelluksen tuottamaa dataa paremmin ja SLAM-teknologiaa voidaan hyödyntää tehokkaammin.

### Merkkipohjainen tunnistus

Merkkipohjaisessa tunnistuksessa sovellus etsii laitteen kameran avulla ennalta määritettyä merkkiä, esimerkiksi kuvaa, kuviota tai kolmiulotteista muotoa. Sovelluksen löydettyä ennalta määritetty merkkikuva, lasketaan kameran koordinaatit, renderoidaan lisätyn todellisuuden malli ja yhdistetään se kameran välittämään

videokuvaan. Merkkipohjainen tunnistus oli tämän työn toteutushetkellä varmatoimisin ja sen vaatiessa vähemmän laskentatehoa kuin esimerkiksi SLAM-seuranta, toimivuus oli parempi myös vanhemmilla laitteilla.

### **Paikannukseen perustava seuranta**

Paikannukseen perustuvassa seurannassa käytetään hyväksi laitteen paikanusominaisuuksia, esimerkiksi GPS-paikannusta, kompassia tai laitteen nopeus- ja kiihtyvyyssantureita. Paikannuksen avulla tehtävä seuranta on toimintavarma GPS-seurannan ansiosta, mutta toimii vain ulkotiloissa ja tekniikalla saatava tarkkuus riippuu käytettävästä laitteesta. Tätä tekniikkaa voidaan kuitenkin hyödyntää esimerkiksi navigointiohjeissa ja lähellä sijaitsevien kohteiden löytämisessä.

## **2.6 Laitteet**

Lisätyn todellisuuden yleisimpiin käytettyihin laitteisiin lukeutuvat mobiili-, päässä pidettävät-, sekä projisoivat-laitteet. Mobiililaitteiden suuren määrän ja nopean kehityksen vuoksi ne ovatkin saaneet suurimman osan sovelluskehityksestä, projisoivien laitteiden ollessa vielä vähäisimmällä käytöllä. Päässä pidettävät laitteistot ovat vielä hinnaltaan kalliita, joten niiden läpimurtoa joudutaan vielä odottamaan.

### **2.6.1 Päälaitteet**

Päässä pidettävät laitteet voidaan jakaa kahteen kategoriaan video- ja optisiin näyttöihin. Kummassakin tapauksessa käyttäjän näkemään kuvaan on lisätty elementtejä. Päälaitteissa on lähes poikkeuksetta myös kaiuttimet äänentoistoa varten, näiden kuitenkin estämättä kuulemasta reaailmaailman ääniä.

## Optiset näytöt



Kuva 5. Microsoft HoloLens [17].

Optisia näyttöjä käyttävä henkilö näkee todellisen maailman näytön läpi, sekä samalla myös näytöllä näkyvän syötteen. Esimerkkinä voidaan mainita vuoden 2016 alkupuolella julkaistu Microsoft HoloLens (kuva 5), jossa käyttäjän näkökenttään heijastuu hologrammeja todellisen maailman näkymien päälle. Hologrammit näkyvät hyvin pienessä osassa näkökenttää, mikä rajoittaa hieman lasien käyttömukavuutta. Optisten näyttöjen ongelmana voitaisiin pitää myös ympäröivän valaistuksen vaikutusta näkymään, joka liian kirkkaana tai himmeänä ollessaan vaikeuttaa kohteen näkemistä.

## Videonäytöt



Kuva 6. Oculus Rift [18].

Videonäyttöjen avulla myös todellisen maailman näkymät toistetaan käyttäjälle videokuvan avulla. Videonäytön (kuva 6) avulla todelliseen maailmaan lisätystä materiaalista saadaan realistisemmin ympäristöönsä sopiva kuva. Haittapuolena reaali maailman syvyysnäkyminen voi olla puutteellinen välitettäessä kuvaa kummallekin silmälle omalla kamerallaan.

### 2.6.2 Mobiililaitteet

Mobiililaitteita voivat olla vaikkapa matkapuhelin tai taulutietokone, joissa soveluksen toteuttama syöte käyttäjälle muodostuu laitteen kameran välittämästä

todellisen maailman kuvasta, sekä sovelluksen videokuvaan lisäämistä elementteistä (Kuva 7). Mobiililaitteiden soveltuvuutta lisätyn todellisuuden alustaksi puoltaa monikin asia, esimerkiksi niiden liikuteltavuus, liikettä tunnistavat anturit, tehokas suoritin, että grafiikkaprosessori, GPS, yhteys internetiin, sekä tietenkin kamera. Käytettävien laitteiden tekniikan ja ominaisuuksien kehittyessä, mobiililaitteiden käyttö lisätyn todellisuuden sovellusalustoina on nostamassa suosiota.



Kuva 7. Vuforia [19].

### 2.6.3 Projisoivat laitteet

Laitteen välittämät kuvat projisioidaan käyttäjälle samaan tapaan kuin esimerkiksi valkokankaalle näytetään elokuvaa. Projisoivien laitteiden avulla käyttäjä ei tarvitse erillisiä laseja nähdäkseen lisättyä todellisuutta.

Kehitysasteella oleva, vuonna 2016 perustettu Hololamp (Kuva 8) muodostaa hologrammin halutulle pinnalle. Laitte kartoittaa pinnan, ja seuraamalla käyttäjän kasvojen sijaintia muodostaa hologrammin avulla 3D-kuvan. Se jäljittelee tapaa jolla ihmiset havaitsevat kolmiulotteisia kohteita, antaen vaikutelman oikeasta objektista. [20.]



Kuva 8. Hololamp [20].

### 3 Toimeksianto

#### 3.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi pörssinoteerattu perheyritys Tulikivi Oyj, joka on maailman suurin lämpöä varaavien tulisijojen valmistaja. Konserniin kuuluvat emoyhtiö Tulikivi Oyj, Tulikivi U.S. Inc ja OOO Tulikivi. Yrityksellä työskentelee noin 200 ihmistä. [21, 4] Yhtiön tehtaat sijaitsevat Juuassa, jossa myös pääkonttori sijaitsee, sekä Suomussalmella, Heinävedellä ja Espoossa [21, 102].

Tulikivellä on kolme tuoteryhmää: Tulisijat, Sauna ja Sisustus ja Suomessa yritys onkin markkinajohtaja varaavissa takoissa. Asiakaskunta koostuu pääasiassa uudisrakentajista ja remontoijista, jotka arvostavat bioenergiaa lämmitysmuotona sekä puulämmityksen taloudellisia hyötyjä. [21, 9.]

### 3.2 Tavoitteet

Toimeksiannon tavoitteena oli tutkia lisätyn todellisuuden mahdollisuuksia tuotemalliston markkinointia ajatellen. Sekä kehittää tähän tarkoitukseen soveltuva mobiilisovellus toimeksiantajan tuotemalliston esittelyyn.

Vuoden 2017 vuosikertomuksessa mainitaan yrityksen hakevan kasvua ja kannattavuutta kohderyhmää laajentamalla; on kaksi eri kohderyhmää sekä modernit että perinteiset kuluttajat. [21, 15.] Lisätyn todellisuuden sovelluksella voitaisiin houkutella ja tehdä vaikutus tehokkaammin etenkin modernimpaan kohderyhmään, tuomalla nykyaikainen tekniikka markkinoinnin avuksi.

Toimeksiantajan suunnitteluohjelmistolla luodun mallin siirtäminen lisätyn todellisuuden sovelluksen käyttöön vaati selvittämistä ja työn sisältämät tekniset välinevalinnat, sekä toimintaprosessit oli dokumentoitava. Sovelluksessa käytettävän mallin seuranta ja mallinnus suhteessa ympäristöönsä vaati aiheeseen perehtymistä. Myös mallin visuaalinen näyttävyys ja oikeat mittasuhteet olivat yksi työn tärkeimmistä kriteereistä. Lisäksi luotavan lisätyn todellisuuden sovelluksen käyttöliittymän tuli olla helppokäyttöinen. [22;23.]

### 3.3 Lähtökohdat

Opinnäytetyön lähtökohtana oli SolidWorks-suunnitteluohjelmistolla toteutettu kokoonpanopiirustus toimeksiantajan Karelia-malliston Raita-takasta [24]. Tämä malli oli tarkoitus siirtää ja valmistella lisätyn todellisuuden avulla esitettäväksi kohteeksi. SolidWorks-suunnitteluohjelmistolla suunnitellut mallit oli tehty tuotantoa ajatellen, joten työn aluksi oli tutkittava mahdollisuuksia mallin keventämiseksi ja visuaalisuuden parantamiseksi enemmän markkinointikäyttöön soveltuvaksi malliksi. SolidWorks-ohjelmistosta ulos saatavat tiedostotyypit eivät olleet Unity-yhteensopivia, näin ollen työhön liittyi myös eri mahdollisuuksien selvittäminen mallien siirtämiseksi Unity-pelimoottoriin sopiviksi ja toimiviksi kohteiksi.

Tekijällä oli aiempien projektien ansiosta kokemusta lisätyn todellisuuden sovelluksien toteutuksesta Unity-pelimoottorilla, sekä näiden saattamisesta myös Android-käyttöjärjestelmälle sopiviksi. Lisätyn todellisuuden sovelluskehityksessä oli etsittävä ratkaisuja tilan mallinnukseen tai vastaavantyyppisen paikannuksen toteutukseen, jolla kohde voitaisiin saada halutulle paikalle. Selvityksen alla oli myös Unity-pelimoottorin kirjastojen, lisäosien ja liitännäisten vertailu, sekä valinta näiden väliltä joita voitaisiin käyttää tässä työssä hyväksi. Sovelluksen alkuvalmistelut ja toteutustyö voitiin aloittaa jo ennen varsinaisen mallin tuomista sovellukseen.

Tämän työn aikana tehty selvitys- ja tutkimustyö tehtiin pääasiassa tutustumalla kulloinkin kyseessä olevien ohjelmistojen ja laitteiden omiin dokumentointeihin, sekä näiden olemassa oleviin sovelluskehittäjälle tarkoitettuihin materiaaleihin.

## **4 Toteutustavan valinta**

Aiempi kokemus Android-käyttöjärjestelmälle toteutettavista sovelluksista, sekä työhön käytettävissä olevat testilaitteet vaikuttivat kyseisen käyttöjärjestelmän valintaan sovelluskehityksen alustaksi. Androidin ollessa mobiililaitteiden keskuudessa laajimman laitepohjan omaava käyttöjärjestelmä ja näin ollen sovelluksen toimivuutta tulevaisuuden näkökulmasta ajatellen, alusta oli luonnollinen valinta. Sovelluksen Android-käännöstä varten tarvittiin Android-sovelluskehityspaketti liitettäväksi Unity-pelimoottoriin, jolla ohjelman lopullinen käännös toteutettiin.

Unity-pelimoottori oli projektiin parhaiten sopiva kehitysalusta, tutkinnan alla oli myös muita vaihtoehtoja. Unity-pelimoottorin puolesta kuitenkin puhui moni asia; työn toteuttajan aiempi kokemus, tiedonsaanti, ohjeiden ja dokumentaatioiden vaivaton saatavuus, koko ajan kehittyvä alusta, sekä helposti tarjolla olevat lisäosat ja laajennukset.

Vuforia-sovelluskehitysalusta, joka työn aloitus- ja toteutushetkellä oli soveltuvim vaihtoehto eri valmistajien alustoille ja käyttöjärjestelmille, sekä näiden eri

versioille, valikoitui lisätyn todellisuuden osion toteutukseen. Yhteensopivuus Unity-pelimoottorin kanssa, hyvin saatavilla olevat ja kattavat dokumentaatiot, sekä kehitysalustan ilmainen käyttömahdollisuus olivat suurimpia vaikuttimia alustan valinnassa.

Ensimmäisenä selvitystyön alla oli myös, millaisia tiedostomuotoja olisi mahdollista tuoda ulos SolidWorks-suunnitteluohjelmistosta. Sekä mitkä näistä tiedostotyypeistä soveltuisivat parhaiten vietäväksi Unity-pelimoottoriin ja millaisia toimenpiteitä tai muutoksia tämän prosessin läpivienti vaatisi.

#### **4.1 Unity-pelimoottori**

Unity Technologiesin valmistama järjestelmäriippumaton pelimoottori, jonka ensimmäinen versio julkaistiin vuonna 2005. Pelimoottori tukee sekä 2D- että 3D-grafiikkaa, ääni- ja videotiedostoja, vedä ja pudota -toimintoa, sekä ohjelmointia komentosarjakielen avulla. Nykyään Unity tukee ainoastaan ohjelmointikieltä C#. Aiemmin ohjelmointi oli myös mahdollista JavaScript-kielellä, joka poistui käytöstä Unity versiossa 2017.1, sekä Unity 5 versiossa poistuneella Boo-kielellä [25].

Unity-pelimoottorin avulla voidaan kehittää sisältöä lähes mille tahansa laitteelle, tuen löytyessä 28 eri alustalle. Sovelluksella voidaan toteuttaa niin käyttöliittymät, kuin itse sovelluksetkin, niin työpöytä-, konsoli-, verkko-, mobiili-, kuin älytelevisio-sovelluksinakin. Unity on erittäin laaja ja toimiva työkalu pelien, että myöskin lisätyn todellisuuden sovellusten kehitykseen, sen tukiessa viimeisimpiä lisätyn todellisuuden teknologioita, muun muassa Vuforiaa, Googlen ARCorea, sekä Applen ARKitä. [26.]



#### **4.1.1 Unity-lisenssi**

Kaikkiin Unity-versioihin on saatavilla tulevat päivitykset, versiot ovat vapaita rojaltimaksuista, sekä kaikki laitealustat ovat vapaasti käytössä. Pelimoottorin ammattimaiseen käyttöön tarvitaan kuitenkin jokin seuraavista lisensseistä, riippuen yrityksen koosta. [27.]

##### **Personal-lisenssi**

Hinta: Ilmainen

Voidaan käyttää, mikäli yrityksen vuotuinen liikevaihto on alle 100.000 dollaria. Suunnattu enemmänkin harraste- ja tutustumiskäyttöön.

##### **Plus-lisenssi**

Hinta: 35 \$ / kuukausi

Mikäli yrityksen vuotuinen liikevaihto on 100.000-200.000 dollaria.

Plus-versio tuo seuraavia lisäominaisuuksia

- Unity Game Dev -kurssit
- 20 %:n alennus Asset Storen tuotteista
- Bolt -visual scripting -työkalu, jolla voidaan ohjelmoida visuaalisesti.

##### **Pro-lisenssi**

Hinta 125 \$ / kuukausi

Yrityksen vuotuiset tulot ylittävät 200.000 dollaria, ei ylärajaa.

- Plus lisenssitason ominaisuudet
- Data analyysit
- Lähdekoodin lisensiointi
- Premium tason tuki.

## Enterprise

Mikäli sovelluskehittäjiä on työryhmässä yli 20, tarvitaan Enterprise-lisenssi. Sisältää Plus ja Pro-lisenssitason ominaisuudet ja ratkaisuja voidaan mukauttaa sopimuksen mukaan. Lisenssistä on sovittava tapauskohtaisesti ohjelmiston toimittajan kanssa.

### 4.1.2 Unity-käsitteitä

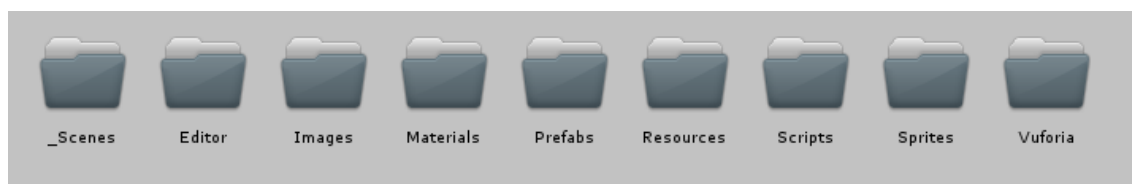
Tässä työssä esitellään niitä Unity-pelimoottorin ominaisuuksia, joita on tarvittu tämän työn toteutuksessa. Muita ominaisuuksia, joita löytyy runsaasti myös muihin käyttötarkoituksiin, ei käydä läpi tämän työn dokumentoinnissa.

#### Assets-kansio

Luodun projektin Assets-kansio on paikka, jonne tallennetaan kaikki projektiin liittyvät tiedostot. Unity-pelimoottori osaa hakea kansiota ja sen alikansioista tehdyt muutokset automaattisesti järjestelmään.

#### Hakemistorakenne

Käyttäjä voi itse määritellä hakemistorakenteet, mukaan lukien niiden, että objektien nimeämiset. Tiedostoja voi lisätä Unity-pelimoottoriin, sekä siirrellä sen sisällä vapaasti, ohjelman hoitaessa uudelleen reititykset ja tarvittavat tiedostomuutokset itsenäisesti. Kehittäjän näkökulmasta on kuitenkin hyvä pitää yllä selkeää hakemistorakennetta ja nimeämiskäytäntöä (kuva 9).



Kuva 9. Unity hakemistorakenne (Kuva: Jukka Pirinen)

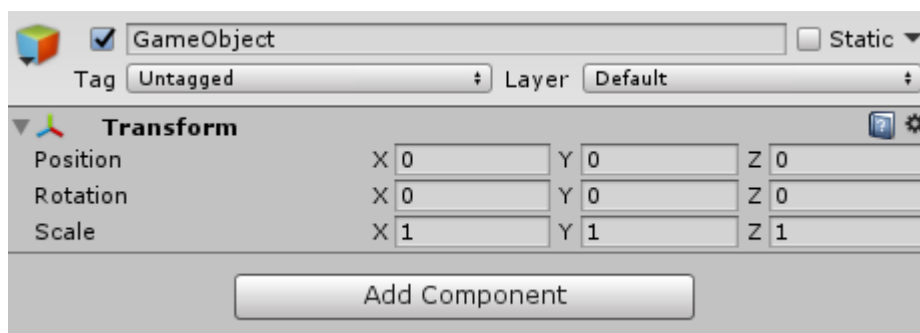
## Scene

Scene eli kohtaus, näkymä, sisältää sovelluksen eri ympäristöt ja valikot. Kohtausten avulla saadaan pidettyä sovelluksen eri osa-alueet paremmin järjestyksessä. Sovellukseen tarvitaan vähintäänkin yksi kohtaus ja eri kohtaukset lisätään sovelluksen kääntämisvaiheessa tehtävään sovelluspakettiin mukaan. Sovelluksen eri osiot sijoitetaan omiin näkymiinsä. Ensimmäisenä kohtauksena sovelluksen käynnistyessä on hyvä olla esimerkiksi alkuvalikko-näkymä, josta käyttäjän on helppo lähteä liikkeelle.

Tässä opinnäytetyössä toteutetussa sovelluksessa käytettiin kahta eri kohtausta. App-kohtaus jossa varsinaisen lisätyn todellisuuden sovelluksen toiminnallisuus sijaitsee, sekä Menu-kohtaus jonka sisältä löytyy alkuvalikko toimintoihin.

## GameObject

Gameobject eli peliobjekti on Unityn peruskomponentti. Kaikki muut käytettävät objektit liitetään luomisvaiheessaan peliobjektiin ja näin ollen se voi pitää sisällään erityyppisiä ominaisuuksia joilla sovelluksen toiminnallisuus on toteutettu. Unity liittää automaattisesti sovellukseen lisättävien objektien pohjaksi peliobjektin. Perusmuodossaan objektilla on ainoastaan Transform-ominaisuus (kuva 10), joka määrittää objektin sijainnin, kierron ja skaalauksen. Perusmuotoista peliobjektia voidaan käyttää myös esimerkiksi näkymättömänä ankkuripisteenä muille objekteille, jolloin näille saadaan lisättyä erilaisia toiminnallisuksia helpommin.



Kuva 10. GameObject, peliobjekti (Kuva: Jukka Pirinen)

## **Materials**

Unity-pelimoottorin materiaalit määrittelevät miten pinta tulee renderoida, sisällyttämällä viittaukset muun muassa laatoituksesta ja värisävyistä käyttämiinsä tekstuureihin. Materiaalin käytettävissä olevat vaihtoehdot riippuvat siitä millaista varjostusta materiaali käyttää. [28.]

## **Shader**

Käytetään sovelluksessa varjostusten toteutukseen ja kohteisiin saadaan näin luotua realismia. Varjostukset ovat pieniä komentosarjoja, jotka sisältävät matemaattiset laskelmat ja algoritmit kunkin pikselin värin laskemiseksi, perustuen valontuloon ja materiaalien asetuksiin [28].

## **Textures**

Tekstuurit ovat bittikarttakuvia. Materiaali voi sisältää viittauksia tekstuureihin, jotta materiaalin varjostus voi käyttää tekstuuria laskettaessa peliobjektin pinnan värin. Peliobjektin pinnan perusvärin lisäksi tekstuurit voivat edustaa monia muita materiaalin pinnan näkökohtia, kuten heijastavuutta tai karheutta. [28.]

## **Particle Systems**

Partikkeleita käytetään Unity-pelimoottorissa erityisesti nesteiden, savun, pilvien ja liekkien luomiseen. Partikkelilla saadaan aikaan luonnollista juoksevuutta ja energian liikettä paremmin esille kuin normaaleilla objekteilla luotuna. Yksittäinen partikkeli on kevyesti toteutettu, joten näitä voidaan renderoida suuria määriä kerrallaan ja näin ollen saadaan aikaan pehmeämpiä efektejä.

## **Script**

Komentosarja, skripti on lyhyt tietokoneen ohjelma, jolla voidaan luoda moninaisia toiminnallisuksia sovellukseen. Komentosarjoille kannattaa luoda oma Scripts-kansio, jonne tehdyt skriptit sijoitetaan ja josta ne ovat helposti

löydettävissä. Unity luo skripti-pohjan automaattisesti, sisältäen oletuksena funktiot `Start()` ja `Update()`.

### **Start()-funktio**

`Start()`-funktioita kutsutaan automaattisesti kerran suoritettavan kohtauksen käynnistyessä ja se suoritetaan ennen `Update()`-funktioita. Tätä funktiota voidaan ajatella luokan konstruktorina.

### **Update()-funktio**

Unity kutsuu automaattisesti `Update()`-funktioita jokaisen kuvan päivityksen aikana, eli funktiota kutsutaan koko sovelluksen ajan ajan. Tämän funktion sisällä ei suositella tehtäväksi raskaita laskutoimituksia, sen hidastaessa näin ollen ohjelman suoritusta, koska seuraavaa päivitettävää kuvaa voidaan kutsua vasta edellisen latauduttua kokonaisuudessaan. Päivitystaaajuuden ollessa esimerkiksi 60 Hz, funktion sisältöä kutsutaan 60 kertaa sekunnissa koko ohjelman suorituksen ajan.

#### **4.1.3 3D-formaatit**

Unity itsessään ei sisällä mallinnustyökaluja, lukuun ottamatta joitakin Asset Store-kaupan laajennuksia, mutta pelimoottoriin voidaan kuitenkin tuoda useampia 3D-mallinnuspaketteja. Unity voi lukea 3D-formaattia olevia fbx, dae (Collada), 3ds, dxf, obj ja skp -päätteisiä tiedostomuotoja. Myös seuraavia kolmannen osapuolen alkuperäistiedostomuotoja on tuettu: Max, Maya, Blender, Cinema4D, Modo, Lightwave ja Cheetah3D.

FBX-tiedostomuoto on kuitenkin käytännöllisin ja suositelluin 3D-objektien tuontiin, sen tukeessa myös animaation tuonnin. Lisäksi Unity kääntää automaattisesti alkuperäistiedostomuodossa tuodut kohteet FBX-muotoon.

Lyhyesti sanottuna, sinun tulisi lähestulkoon aina käyttää FBX-tiedostomuotoa. Tämä on Unity-yhteisön kaikkein laajimmin tuettu, käytetty ja testattu formaatti ja se tukee kaikentyyppisten, staattisten sekä animoitujen polygonimallien tuontia; antaen parhaan tuloksen. [29, 24.]

Alkuperäistiedostomuotoja käyttämällä malliin tulee usein mukaan ylimääräisiä 3D-objekteja, joita ei välttämättä edes tarvita sovelluksessa. Etenkin testausvaiheessa mukana ollut Blender 3D-grafiikan mallinnussovellus lisäsi malliin omia lisäelementtejään, jotka eivät vaikuttaneet visuaalisuuteen, mutta mallin tiedostokoko kasvoi miltei kaksinkertaiseksi. Näiden löytäminen ja poistaminen, joko mallinnusohjelmassa tai Unity-pelimoottorissa olisi tuonut yhden ylimääräisen työvaiheen prosessiin lisää. Tämän lisäksi mallit ovat automaattisesti sidottuna mallinnusohjelmaan josta tuonti tapahtuu. Alkuperäistiedostomuotoja käyttämällä tarvitaan myös kyseisen ohjelmiston täydet lisenssit jokaiselle laitteella, jolla projekti aiotaan avata [29, 14]. Hyötynä alkuperäistiedostomuotojen käytössä, kohteen muokkaus mallinnusohjelmassa tuo muutokset myös Unity-pelimoottoriin tuotuun malliin.

## 4.2 Google ARCore

Googlen kehittämä, Unity-tuettu lisätyn todellisuuden ohjelmistokehityspaketti ARCore käyttää virtuaalisen ja todellisen maailman sisällön yhdistämisessä liikkeen tunnistusta, jonka avulla laite tunnistaa sijaintiaan suhteessa reaali maailmaan. Alustan avulla sovellusta käyttävä laite voi tunnistaa pintojen kokoja ja sijainteja. Myös ympäröivän todellisen maailman valon huomioon ottaminen sovelluksen ajon aikana on mahdollista. [30.]

Koska ARCore käyttää pintojen tunnistuksessa pisteiden havaitsemista, voi liian tasaisten ja sileiden pintojen tunnistamisessa olla ongelmia. Toisaalta tämän

ominaisuuden ansiosta pintojen ei myöskään tarvitse olla tasaisia tai vaakasuorassa olevia.

Google kehitti myös Tango-projektia, joka oli aiempi lisätyn todellisuuden ohjelmistokehityspaketti. Tällä tekniikalla toteutetut sovellukset olivat laitteistojen osalta rajoitetumpia, niiden vaatiessa käyttävältä laitteeltaan muun muassa syvyyskameraa. Google lopetti tämän projektin kehityksen vuoden 2018 alkupuolella ja keskittyy nykyään ainoastaan ARCore-alustan kehitystyöhön [31; 32].

ARCore toimi työn toteutuksen aikana vain version N (7.0) tai sitä uudemmissa Android-käyttöjärjestelmän omaavissa laitteissa. Työn aloitusvaiheessa paketti oli vielä kehitysvaiheessaan ja siitä ei ollut saatavilla vakaata versiota. Tämä seikka, sekä sovellusta käyttämään pystyvien laitteiden rajattu määrä, johtivat kehityspaketin hylkäämiseen tämän työn toteutusta ajatellen.



Kuva 11. ARCore-kehitystyökalulla toteutettu sovellus (Kuva: Jukka Pirinen)

### 4.3 Apple ARKit

Applen luoma lisätyn todellisuuden ohjelmistokehitystyökalu, jolla on tuki myös Unity-pelimoottorille. Työkalulla luodut sovellukset toimivat iOS 11 tai sitä uudemmissa käyttöjärjestelmissä tai mikäli laite on valmistettu vuoden 2015 jälkeen.

ARKit-alustalla on tuki muun muassa seuraaville ominaisuuksille: laitteen liikkeen seurannalle, paikannukselle, sekä tilan ymmärtämiselle, jolla yhdistetään laitteen kameran välittämää tietoa tietokoneen tekemään visioanalyysiin. Alusta käyttää kameran välittämässä pintojen tunnistuksessa Applen omaa TrueDepth-kameraa, jolla voidaan havainnoida myös kuvassa näkyvää syvyyttä. Alusta käyttää myös hyväkseen valaistuksen tunnistusta, jolla voidaan säätää virtuaalisesti luotuun objektiin osuvan valon määrää, riippuen todellisen maailman valosta. [33.]

ARKit-ohjelmistotyökalun avulla kehitettyjen sovellusten käytön rajoituessa kuitenkin vain Applen omien ja uusimpien laitteiden käyttöön, jouduttiin tämän kehitysalustan käyttö hylkäämään tässä opinnäytetyössä.

### 4.4 Vuforia AR SDK

Vuforia on lisätyn todellisuuden sovelluskehityspaketti, joka on käytettävissä Unity-, Android- ja iOS-kehitysalustoilla. Saatavilla on myös kehityspaketti Windows 10 laitteille, sekä HoloLens-älylaseille.

Vuforia sovelluskehityspaketissa on saatavilla erilaisia toimintoja aina 3D-objektien tunnistuksesta, kuvan- ja tekstintunnistukseen, myös pilvipalvelun ja laitekohtaisen tunnistuksen hyödyntäminen on tuettu. Alustalla on myös mahdollista toteuttaa virtuaalisia painikkeita, sekä näyttää videokuvaa havaitulla kohdepinnalla.

Vuforia käyttää tietokantanaan pilvipalvelua, jonne merkkinä käytettävä kuva tai objekti on tallennettava ennen sovelluskäyttöä. Myös sovelluskohtainen



ohjelmointirajanpinnan avain, eli lisenssiavain täytyy rekisteröidä ja noutaa pilvipalvelusta.

Tämän työn toteutuksessa käytettiin kuvantunnistus-ominaisuutta, sen ollessa soveltuvien ja varmatoimisten useiden eri alustojen käyttöön toteutushetkellään. Vuforia kuvantunnistus havaitsee ja seuraa kuvasta löytyviä ominaisuuksia ja vertaa näitä tietokannasta löytyvään kuvaan [34].

## **Ground Plane**

Vuforia Ground Plane -toiminto mahdollistaa digitaalisen sisällön sijoittamisen vaakasuoralle pinnalle, kuten lattialle tai pöydälle ilman merkkikuvaa [35]. Toimintaperiaate on käytännössä samantapainen kuin ARKit ja ARCore ohjelmistokehitystyökaluilla toteutetuissa sovelluksissa.

Opinnäytetyön alkuvaiheessa tämä toiminto oli saatavilla vain testauskäyttöön tietyille iOS-käyttöjärjestelmän laitteille ja toimintoa ei ollut ollenkaan saatavilla Android-laitteille. Opinnäytetyön viime hetkillä ja sovelluksen ollessa jo käytännössä valmiina, Vuforia julkaisi päivityksen, joka mahdollisti ominaisuuden käyttöön laajalla määrällä eri mobiililaitteita. Tästä ominaisuudesta ei tosin ollut enää hyötyä tässä työssä toteutettavassa sovelluksessa.

Toiminto on kuitenkin jatkokehitystä ajatellen varteen otettava vaihtoehto merkittömälle tunnistukselle. Unity Ground Planen tuessa laajaa joukkoa mobiililaitteita ja Unityn alustasitoutumattoman kehityksen näkökulmasta ajatellen, se on myös käyttökelpoinen vaihtoehto tulevaisuudessa ARKit ja ARCore alustoille.

## **Lisenssi**

Vuforian ilmainen lisenssi oli riittävä tähän projektiin. Vähintäänkin Classic-lisensitasoa olisi tarvittu, mikäli olisi haluttu esimerkiksi käyttää tietyn muotoisen objektin tunnistusta tai 3D-skannausta, jotka eivät olleet tarpeen tässä työssä. Mikäli sovelluksessa käytetään kuvantunnistusta suoraan pilvipalvelusta, on käyttökerrat rajattu kuukaudessa 1000 tunnistukseen, tämän ylittyessä on tarkasteltava eri lisenssivaihtoehtoja. Tämän työn tapauksessa tietokantaan

lisätty merkkikuva-paketti ladattiin sivustolta sovelluksen kehitysvaiheessa mukaan ja kuvantunnistus tehtiin itse valmiissa sovelluksessa.

Vuforia lisenssivaihtoehdot: Free (Ilmainen), Classic (499 \$, kertamaksu), Cloud (99 \$ / kuukausi), Pro (Sopimuksen mukaan) [36].

## **4.5 SolidWorks**

Toimeksiantaja käyttää mallistonsa suunnittelussa SolidWorks 3D CAD-suunnitteluohjelmistoa. Valmistajan omien sivustojen mukaan ohjelmisto tarjoaa valikoiman 3D-työkaluja, joiden avulla voi luoda, simuloida, julkaista ja hallita tietoja [1]. SolidWorks-ohjelmistoa on laajennettavissa käytettävyyden lisäämiseksi erikseen saatavilla lisäosilla ja liitännäisillä, niin alkuperäisvalmistajan kuin kolmansien osapuolien tahoilta.

### **Tiedostomuodot**

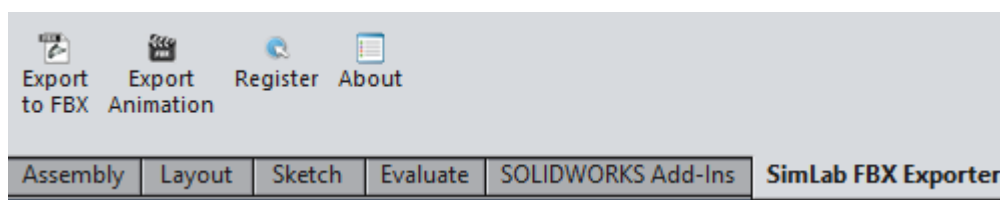
SolidWorks käyttää alkuperäismuodossaan olevien komponenttien tallentamiseen seuraavia eri tiedostomuotoja.

- SLDPRT, Part Document, yksittäisen osan tallennusmuoto
- SLDASM, Assembly Document, kokoonpanotiedosto, koostuu yksittäisistä osista
- SLDDRW, Drawing File, valmistuspiirustus.

### **FBX Exporter -liitännäinen**

Mallin käännös FBX-tiedostomuotoon ja tätä myötä Unity-yhteensopivaksi, toteutettiin SimLab Soft -yhtiön tarjoamalla FBX Exporter SolidWorks -liitännäisellä (kuva 12). Liitännäinen on suunniteltu juuri FBX muunnoksiin virtuaalitodellisuuden käyttöä ajatellen [37].

Liitännäinen vaati rekisteröitymisen, jonka jälkeen käytössä oli ilmainen kokeiluversio. Kokeiluversio oli toiminnassa joko 30 käännöksen verran tai kahden viikon ajanjakson, riippuen siitä kumpi täyttyi ensin. SimLab Soft tarjoaa täyttä versiota liitännäisestä; yhden käyttäjän henkilökohtaisen, kahdella laitteella käytettävän täyden lisenssiversion saa hintaan 149.00 \$ ja useammalle käyttäjälle suunnatun lisenssin hintaan 299.00 \$.



Kuva 12. FBX Exporter liitännäinen SolidWorks-ohjelmiston valikkopalkissa

## 4.6 3D-mallinnusohjelmat

### Rhinoceros

Rhinoceros 3D-mallinnusohjelmisto pystyy tallentamaan tiedoston fbx-muotoon, joten muutosta koetettiin tämän sovelluksen avulla. Ohjelmiston olisi pitänyt pystyä avaamaan suoraan SolidWorks-suunnitteluohjelmiston alkuperäis-tiedostot, tämä ei kuitenkaan onnistunut. Syyksi paljastui Rhinoceros 5, joka oli liian vanha versio sopiakseen yhteen SolidWorks 2015-ohjelmiston kanssa. Uudempi versio ohjelmasta, Rhinoceros 6, pystyi avaamaan käytössä olleen SolidWorks-tiedoston ja muuntamaan sen fbx-tiedostomuotoon.

### Blender

Blender on ilmainen avoimen lähdekoodin 3D-mallinnusohjelma, jolla pystytään tallentamaan tiedostot fbx-muotoon ja jonka alkuperäistiedostomuotoja Unity hyväksyy. SolidWorks-ohjelmistolla tallennus täytyi tehdä stl-tiedostomuotoon, niiden saamiseksi Blenderiin. Tuodut tiedostot eivät tosin tulleet kokonaisina, stl-muunnoksen johdosta. Lisäksi sovellus kasvatti mallin kokoa myös fbx-

tallennuksessa lisäämällä ainakin kamera- ja valo-objektin, joita ei haluttu mukaan vielä tässä vaiheessa.

## Tulokset

Rhinoceros 6 lisäsi mallin kokoa reilusti ja ohjelmiston ilmaisen kokeiluversion päättyessä, täyden version hinta olisi ollut 995 €. Se ei ollut näin ollen toimivin vaihtoehto muunnoksen toteutukseen. Blenderin jättäessä mallista osia pois ja kasvattaessa turhaan tiedostonkokoja jo puutteellisesti siirtyneestä mallista, myös se joutui hylkäykseen.

Rhinoceros 6 ja Blender-sovelluksen, sekä SimLab FBX Exporter-liitännäisen lopulliset eroavaisuudet (kuva 13) Unity-pelimoottoriin tuotuna fbx-tiedostoina. Rhinoceros 6 ja Blender-ohjelmien tuottamat muunnokset olivat reilusti suurempia vertailussa parhaan tuloksen tuottaneeseen. SimLab FBX-Exporter-liitännäinen toi mallin kolme kertaa pienemmässä koossa Unity-pelimoottoriin, säilyttäen kuitenkin kaikki alkuperäiset komponentit ja lisäämättä mitään ylimääräisiä elementtejä mukaan.

	Blender	Rhino6	SimLab
Objects (pcs)	469	267	162
Meshes (pcs)	244	253	135
GameObjects (pcs)	225	14	27
Size (KB)	29183	30657	10536

Kuva 13. Blender–Rhinoceros 6–SimLab FBX Exporter

## 5 Sovelluksen toteutus

Tuotemallin esittelysovellus toteutettiin Unity-pelimoottorin 3D-projektina, käyttäen Vuforia Unity-ohjelmistokehityspakettia. Sovellukseen toteutettiin kaksi erillistä sovellusnäkymää, aloitusvalikko ja varsinainen lisätyn todellisuuden sovellusnäkymä. Valmistunut sovellus tunnistaa käytettävän merkkikuvan ja renderoi tämän päälle toimeksiantajan mallin luonnollisessa koossaan. Toteutuksen lopullinen sovellus käännettiin Android-käyttöjärjestelmällä toimivaksi apk-paketiksi.

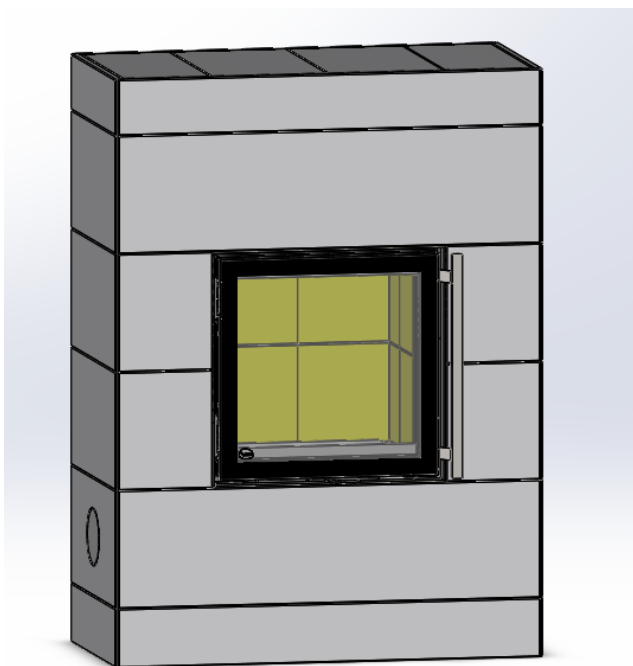
### 5.1 Käytetyt sovellukset ja laitteet

Sovelluksen toteutuksessa käytettiin seuraavia sovelluksia ja lisäosia. Seuraavassa listauksessa mainitaan myös sovelluksen kehityksenaikaisessa testauksessa käytetyt laitteistot ja niiden käyttöjärjestelmät.

- Unity 2017.2.0f3 (64bit)
- Android Studio 3.0.1
- Vuforia 6.5.22
- Microsoft Visual Studio Community 2015, versio 14
- SolidWorks 2015 x64 Edition
- SimLab FBX Exporter
- Sovellustestauksessa: Samsung Galaxy S8, Android 8.0 sekä Sony Xperia Z2, Android 6.0

## 5.2 Mallin siirto

Alkuperäisen SolidWorks-ohjelmistossa olleen mallin (kuva 14) tiedostokoko oli 93.5 MB. Tarkoituksena oli tehdä mallista vähemmän tilaa vievä ja kevennetty versio Unity-pelimoottoria varten, joskin alkuperäistä mallia oli jo hieman kevennetty toimeksiantajan toimesta. Alkuvaiheen kokeiluna oli tallentaa koko SolidWorks-malli SLDPRT-tiedostomuodossa, yksittäisenä osana, jolloin mallista olisi saatu käyttöön vain ulkokuoret. Toiminto onnistui, mutta malliin tarvittiin myös luukun lasin läpi näkyviä osia, jotka jäivät tämän toimenpiteen seurauksena pois. Lisäksi mallin yksittäiset osat eivät enää olisi olleet Unityssä muokattavissa ja toiminnallistettavissa erikseen, joten tämä kyseinen toimintatapa jouduttiin hylkäämään.



Kuva 14. Alkuperäinen malli SolidWorksissa

SolidWorks voi alun perin kääntää omalla export-toiminnollaan tiedostot stl-formaattiin, joka voidaan muuntaa muiden sovellusten avulla Unity yhteensopivaan fbx-muotoon.

Useiden testailujen ja mallin kiertämisen eri mallinnussovellusten kautta, parhaana vaihtoehtona mallin siirtämiseksi fbx-muotoon valikoitui SolidWorks-ohjelmistoon saatavilla oleva SimLab FBX Exporter-liitännäinen. Tämän liitännäisen avulla muunnos haluttuun tiedostomuotoon oli vaivatonta ja mallin tiedostokoko pysyi parhaiten hallinnassa.

## 5.3 Vuforia

Projektissa tarvittava Vuforia-sovelluskehitysalusta oli integroituna Unity-pelimoottoriin tässä työssä käytetystä versiosta alkaen, joten sitä ei ollut tarve ladata erikseen toimittajan sivustolta. Mikäli olisi käytetty Unityn vanhempaa versiota, olisi sovelluskehityspaketti täytynyt ladata editoriin erikseen [38].

### 5.3.1 Vuforia Developer

Vuforia Developer eli sovelluskehittäjä-sivustolla tallennettiin haluttu pohjakuva oman kehittäjätilin tietokantaan niin sanotuksi ”markeriksi” eli merkkikuvaksi. Tämä on kuva, jonka päälle renderoidaan sovelluksen avulla lisätyn todellisuuden malli.

Kuvakaappauksessa (kuva 15) näkyy Vuforia sovelluskehittäjä sivuston tätä työtä varten valittuun merkkikuvaan automaattisesti generoimat pisteet, joiden avulla sovellus tunnistaa ja renderoi lisätyn todellisuuden objektit pohjakuvan päälle. Tarkempi ja yksityiskohtaisempi kuva toisi enemmän tunnistuspisteitä, mutta toisaalta Vuforia-kehittäjä sivusto ei hyväksy kooltaan liian suuria kuvia, maksimi tiedostokoon ollessa 2mb. Kuvaa tallennettaessa sivuston tietokantaan on kuvan leveys määriteltävä, joskaan tällä toiminnolla ei ole muuta merkitystä kuin sovelluksessa käytettävän mittakaavaan skaalautuminen. Kun kuva saatiin tallennettua tietokantaan ja pisteet näyttivät generoituvan riittävän tarkasti, ladattiin haluttu kuva sivustolta Unity-pakettina. Tavoitteena kuvan tasolle olisi saada Augmentable (kuva 15) määrittelyn kohdassa viisi tähteä, huonompi kuvanlaatu voi aiheuttaa ongelmia sovellukselle lukea kuvaa, jolloin lisätyn todellisuuden mallinnus ei toimi halutulla tavalla. Työssä haluttiin käyttää toimeksiantajan logoa merkkikuvana. Testauksien perusteella sovellus näytti toimivan riittävän hyvin myös neljällä tähdellä, joten kuva voitiin hyväksyä myös lopulliseen versioon.

## Tulikivi\_kivi\_logo\_BLACK

[Edit Name](#) [Remove](#)



[Update Target](#) [Hide Features](#)

Type: Single Image

Status: Active

Target ID: 98316f80f6a145fdabce9ccaf6dc7a7a

Augmentable: ★★★★★

Added: Jan 15, 2018 08:46

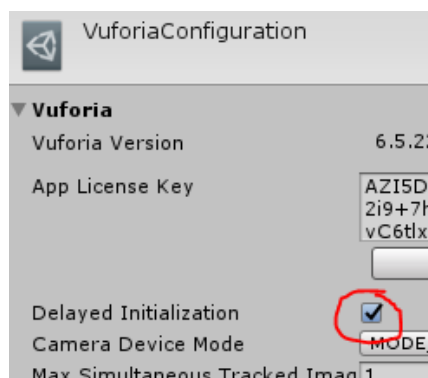
Modified: Jan 15, 2018 08:46

Kuva 15. Merkkikuva Vuforia-kehittäjä sivustolla

### 5.3.2 Vuforia Unity-pelimootorissa

Alkuun Player Settings valikosta täytyi laittaa Vuforia Augmented Reality valituksi, jonka jälkeen Unity lisää Vuforian käyttöön, ja päästään säätämään itse Vuforian-asetuksia. Valinta täytyi laittaa päälle sekä PC-valikosta testausta varten, että Android-valikosta itse käännettävää so-

vellusta varten.

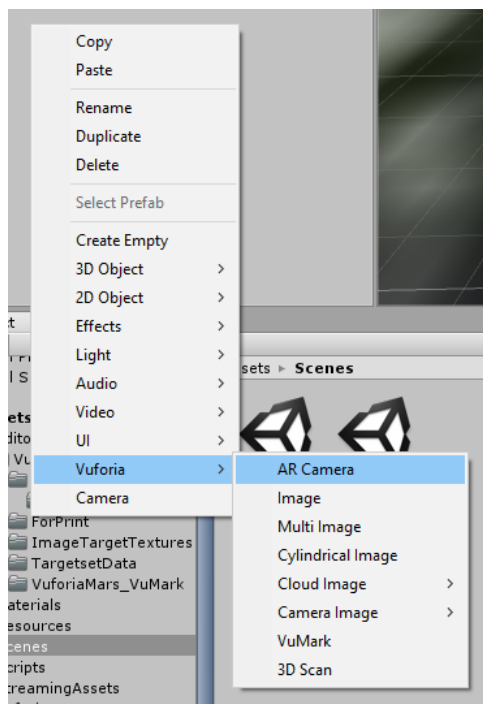


Kuva 16. Viivästetty alustus

Asetuksista kytkettiin Delayed Initialization, eli viivästetty alustus käyttöön (kuva 16). Toiminto estää Vuforian käynnistymisen heti sovelluksen käynnistyessä. Näin ollen sovelluksella voitiin käyttää esimerkiksi aloitusvalikkoa ilman lisätyn todellisuuden osion käynnistymistä. Toimintoa



käytettäessä on Vuforia-alustus käynnistettävä erikseen määritellyllä skriptillä "Liite 3".

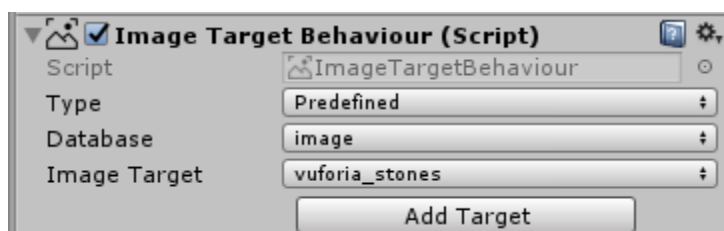


Kuva 17. AR kamera

Kun halutut kehitysalustat oli asennettu, lisättiin sovellukseen Vuforia-valikosta AR-kamera (kuva 17), sekä samasta valikosta löytyvä Image, joka loi Image Target -komponentin sovellukseen. Tämä Image Target -komponentti toimi pohjana, johon liitetään haluttu merkkikuva.

Nämä osat olivat sovelluksen perusta, joten komponenttien sijainnit määritettiin nollaksi. Tällä toimenpiteellä voitiin helpottaa myöhemmässä kehitysvaiheessa lisättävien osien sijoitusta ja toimintojen rakennusta suhteessa edellä mainittuihin osiin.

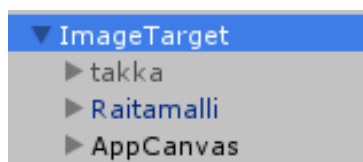
Vuforia-kehittäjä sivustolta ladattu pohjakuvan sisältävä paketti tuotiin pelimootooriin ja kuva liitettiin luotuun Image Target komponenttiin (kuva 18). Näin toteutettuna kuvan tunnistus ja käsittely tapahtuu vain sovelluksessa itsessään, ei pilvipalvelussa.



Kuva 18. Image Target

## 5.4 Malli Unity-pelimoottorissa

SolidWorks-ohjelmistosta fbx-tiedostomuotoon käännetty malli voitiin tuoda pelimoottoriin, joko yksinkertaisesti vetämällä haluttu tiedosto Unity-projektin assets-kansioon tai tuomalla malli import new asset -toiminnolla.



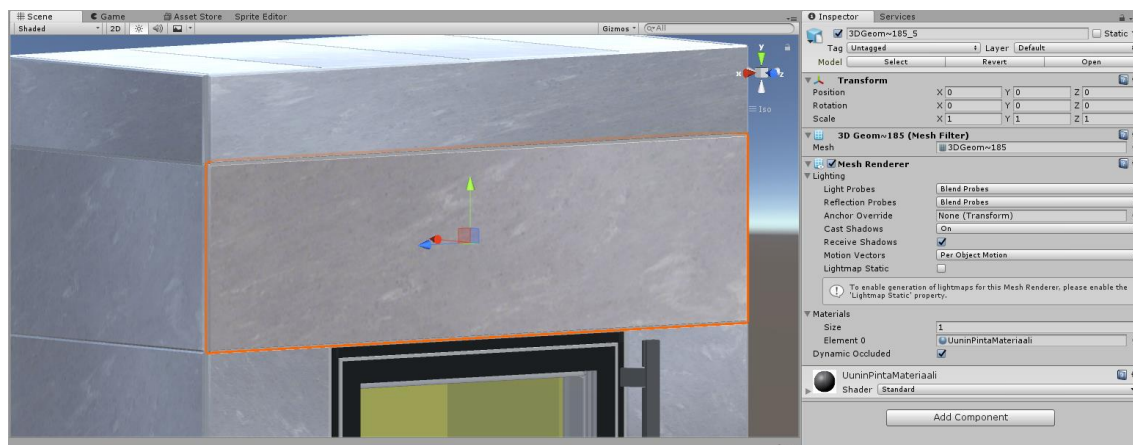
Kuva 19. Lapsi-objektit

Kaikki toiminnot jotka haluttiin toteuttaa lisätyn todellisuuden avulla merkkikuvan ilmestyessä sovelluksen saamaan kuvaan, asetettiin ImageTarget-komponentin lapseksi (Kuva 19), mukaan lukien tuotu malli.

### 5.4.1 Materiaalit

#### Pintamateriaali

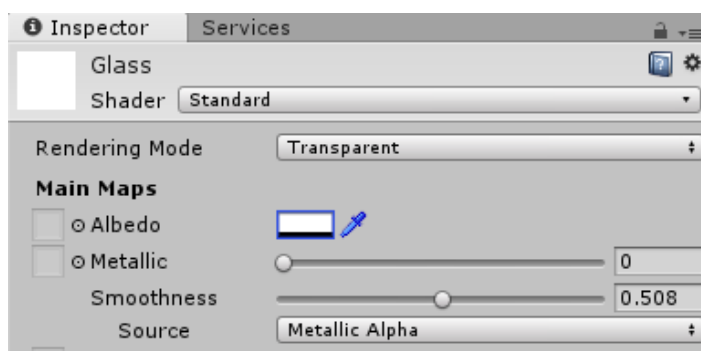
Unity-pelimoottorin tukiessa myös valokuvien tuonnin sovelluskehitykseen, voitiin pintamateriaalia varten tuoda oikean mallin valokuvasta rajattu kuva pintamateriaalista. Materiaalin liittämiseksi mallin pintakomponentteihin luotiin uusi Unity-materiaali, johon tuotu png-tiedostomuotoa oleva kuva voitiin lisätä suoraan (kuva 20).



Kuva 20. Mallin pintamateriaali

## Lasi

Mallin luukussa olevan ikkunamateriaalin pitäisi olla läpinäkyvä, näin ei kuitenkaan ollut mallin siirtyessä Unity-pelimoottoriin. Lasin ominaisuudet puuttuivat jo alkuperäisestä toimeksiantajan toimittamasta mallista, joten läpinäkyvyys oli luotava Unityn omien materiaalelementtien avulla. Luotu materiaali tehtiin läpinäkyväksi asettamalla Rendering Mode -asetus Transparent, eli läpinäkyvään muotoon. Smoothness-säädön avulla voitiin säätää lasin heijastuvuutta (kuva 21). Mallista haettiin luukun osat, joiden oli tarkoitus olla läpinäkyviä ja liitettiin näihin luotu lasimateriaali.

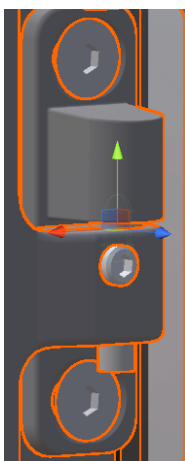


Kuva 21. Lasimateriaali

## Saumat ja tiivisteet

Mallissa ei ollut SolidWorks-mallinnusohjelmasta tullessaan saumoja pintalaattojen väleissä, eikä mallin reunoilla. Myös tiivisteet luukun reunoilta puuttuivat. Nämä luotiin Unity-editorissa käyttämällä 3D-objekteja, jotka liitettiin malliin ja sovitettiin saumojen ja tiivisteiden paikoille sopiviksi.

### 5.4.2 Luukun toiminnallisuus

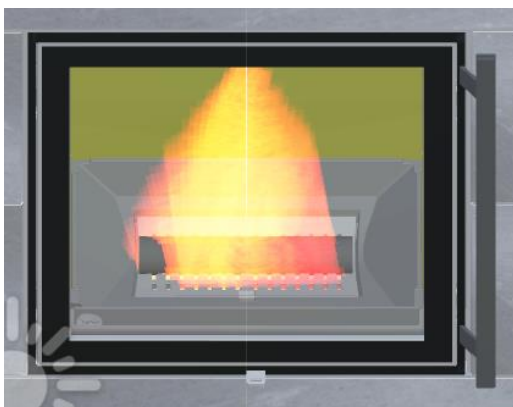


Mallin luukulle lisättiin pieni 3D-sylinteriobjekti saranaksi (kuva 22), jonka lapsiobjekteiksi kaikki luukkuun kuuluvat osat asetettiin. Kiertämällä tätä saranaobjektia pystyakselinsa ympäri, saatiin myös lapsiobjektina oleva luukku kääntymään. Luukun liike ja rajoitettu aukeamiskulma on toteutettu omalla skriptillään (Liite 2). Sovelluksen käynnistyessä luukku on alkuasetuksena lukittuna, mutta toiminnon voi asettaa päälle ja pois asetusvalikon valinnasta.

Kuva 22. Sarana

### 5.4.3 Tuli

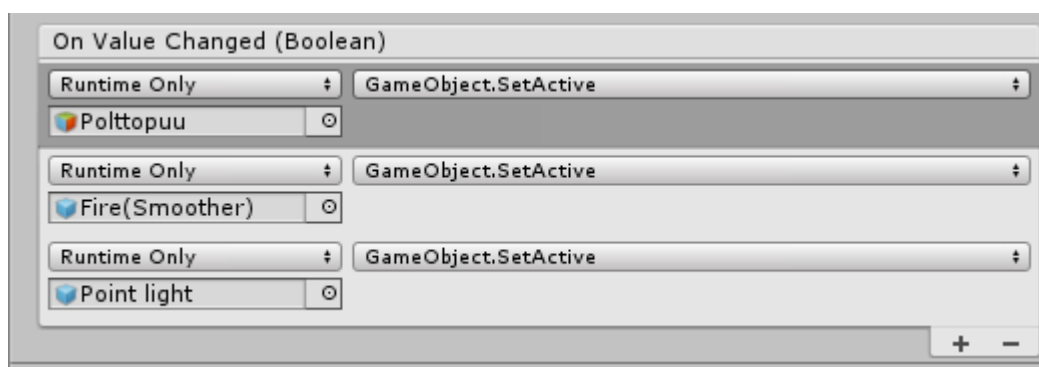
Mallin sisään luotiin liekkejä käyttämällä Particle System -efektiä, sekä polttopuuta kuvaava 3D-sylinteriobjekti (kuva 23). Particle System on tarkoitettu kohteiden, joissa on paljon elementtejä, kuten sateen, höyryn tai vaikka lintuparvien luomiseen. Elementissä on todella paljon säätömahdollisuuksia ja ajan kanssa eri asetuksia säätämällä tulesta saataisiin luotua todella näyttävä.



Kuva 23. Tuli ja polttopuu

Tuli-efekti ei kuitenkaan itsessään luonut valaistusta ympärilleen, joten sille oli lisättävä valo realistisemman ilmeen saavuttamiseksi. Tämän toteutus tapahtui Point Light valo-objektilla, joka sijoitettiin liekin sisään. Valon väriä ja kirkkauksia säädettiin muistuttumaan oikean tulen luomia värejä halutun vaikutelman saavuttamiseksi.

Sovelluksen käynnistyessä puu, tuli ja valo ovat alkuasetuksena poissa näkyvistä, mutta ne voidaan kytkeä päälle ja pois sovelluksen asetusvalikosta. Itse toiminnot tapahtuvat asetusvalikossa sijaitsevan painikkeen On Value Changed -ominaisuuden ja siihen liitettyjen objektien toimintojen avulla (kuva 24).



Kuva 24. Liekkien sammutus

## 5.5 Kohtaukset

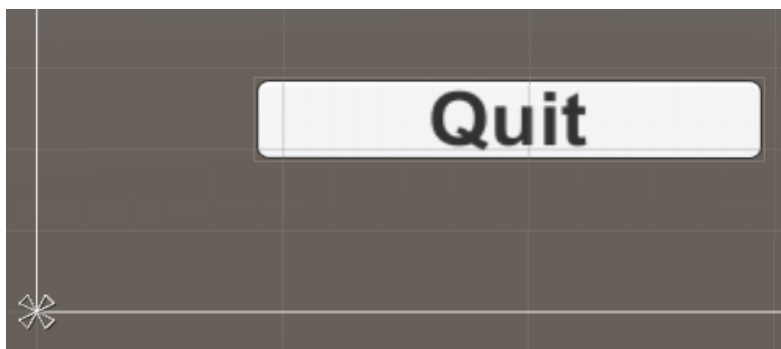
Sovelluksessa on kaksi erillistä kohtausta, sovelluskohtaus, jossa varsinainen lisätyn todellisuuden sovellus toimii, sekä aloitusvalikko eli menu-kohtaus, joka käynnistyy sovelluksessa ensimmäisenä.

Kohtauksissa olevat valikot rakennettiin käyttämällä Unityn UI-Canvas -elementtiä, jonka alla varsinaiset toiminnallisuudet ja painikkeet toimivat. Kohtaukseen lisättyä UI-Canvas -elementtiä on aina kameran käynnistyessä päällimmäisenä ja näin ollen sen avulla on kätevää toteuttaa valikkonäkymät. Valikoiden välillä liikkuminen on toteutettu omilla skripteillään, toiminnot tähän liitteessä 3. Luodut skriptit liitettiin kunkin halutun painikkeen On Click() komponenttiin.

### 5.5.1 Aloitusvalikko

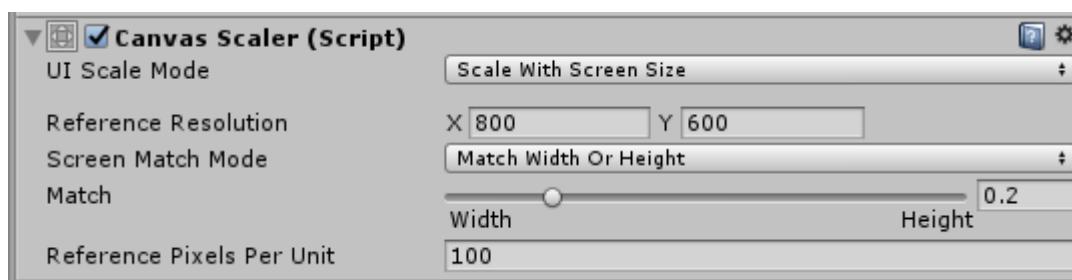
Aloitusvalikkoa varten luotiin sovellukseen oma kohta, Menu-kohta. Menu-kohtaukseen asetettiin käyttöön normaali kamera jolla ei välitetä videokuvaa, kun taas itse sovellus-kohtauksessa käytetään lisätyn todellisuuden kameraa välittämään videokuvaa käyttäjälle. Aloitusvalikko-kohtauksessa on painikkeet sovelluksesta poistumiseen, inforuudun näyttämiseen, sekä painike varsinaiseen sovellukseen pääsemiseksi. Valikoiden rakentamisessa otettiin heti alusta saakka huomioon sovelluksen vaatima responsiivisuus, eli mukautuminen eri kokoisille näytöille.

Painikkeet, sekä muut valikkojen elementit täytyi ankkuroida UI-Canvaksen reunoihin niiden sijainnin varmistamiseksi myös eri kokoisilla näytöillä. Tässä voitiin käyttää apuna UI-komponenteista löytyvää Anchor Preset -toimintoa tai vetämällä manuaalisesti komponentin ankkurointinuolet haluttuun reunaan (kuva 25). Osia voitiin myös ankkuroida toisiinsa, näin saatiin responsiivisuus vielä paremmin hallintaan ja osat kestivät halutulla etäisyydellä toisistaan. Mikäli toimintoja oli näkyvissä enemmän, mukautuvuuden ansiosta kaikki eivät pakkautuneet ruudun pienentyessä yhteen kasaan tai vastaavasti kadonneet käyttäjän näkyviltä.



Kuva 25. Painike ankkuroituna vasempaan alakulmaan

Valikkoelementtien koon skaalaus voitiin toteuttaa Unitystä löytyvällä Canvas Scaler -skriptillä (kuva 26). Canvas Scaler vaatii oikeiden asetusten löytämiseksi hieman säätämistä ja testailuja eri kuvasuhteilla. Komponentista löytyvän Match-ominaisuuden avulla voitiin säätää korkeuden ja leveyden välistä skaalauskerrointa, ja oikean arvon löytäminen vaatii useamman testauskierroksen [39].



Kuva 26. Valikkojen skaalaus

## 5.5.2 Sovellus-kohtaus

### Käyttöliittymät

Sovellus-kohtaukseen on luotu UI-Canvas-elementin avulla kaksi erillistä käyttöliittymää. Ensimmäinen käyttöliittymä on näkyvissä itse sovelluksen käynnistyessä, josta löytyvät painikkeet aloitusvalikkoon palaamiseksi ja asetusvalikkoon siirtymiseen. Toinen käyttöliittymäikkuna tulee näkyviin asetusvalikkoon mentäessä, jossa sijaitsee painike sovellukseen palaamiseksi, sekä kytkimet luukun lukitukseen, että tulen sammutukseen. Myös valaistuksen kirkkauden ja valaistussuuntien säädöt löytyvät asetusvalikosta. Molempien sovellus-kohtauksen käyttöliittymien taustalla on näkyvissä kuitenkin itse sovellus, ja näin ollen asetuksia säätämällä voidaan nähdä vaikutukset lisätyn todellisuuden malliin välittömästi.

### Extended Tracking

Extended Tracking on Vuforia Image Target Behaviour -skriptistä löytyvä toiminto, jolla saadaan lisätyn todellisuuden kohteet pysymään paikallaan vaikka näköyhteys merkkikuvaan olisi menetetty. Kuvan on kuitenkin täytynyt olla alun perin näkyvillä mallin paikalleen renderoimiseksi. Toiminnon avulla myös suurempien mallien tarkastelu onnistuu paremmin.

## Smooth Camera.cs -skripti

Lisätyn todellisuuden malli tärisi sovelluskehityksen alkuvaiheessa hieman koko ajan. Syynä tähän oli sovellusta käyttävän laitteen antureiden ja kameran välisestä yhteistyöstä, jolla sovellus pyrkii ajonaikaisesti laskemaan sijaintia uudelleen pitääkseen mallin juuri oikeassa asennossa ja paikassa. Laite joutuu tekemään sovelluksen ajonaikana uudelleen laskemista mallin pitämiseksi paikoillaan. Tätä toimenpidettä voitiin pehmentää skriptin avulla (Liite 4), joka hidastaa mallin sijainnin uudelleen laskemista.

Uusimpien mobiililaitteiden näytön päivitystaajuus on useimmiten 60Hz [34], kuvan päivityksen viedessä näin ollen 1.67 millisekuntia. Tänä aikana on laskettava uudelleen kameran ja laitteen sijaintia kohteeseen. Tämä voi näin ollen helposti aiheuttaa viivettä sovellukseen, joten turhaa uudelleen laskemista tulisi välttää.

## 5.6 Valo ja varjostukset

### Valo

Mallia valaistaan kahdesta eri suunnasta tulevalla valolla, jotka eivät luo malliin kuitenkaan varjoja. Näillä valoilla oli tarkoitus saada mallin pintamateriaali näyttämään eri tilanteissa mahdollisimman realistiselta. Mikäli näillä valoilla olisi luotu varjot, ne olisivat olleet malliin nähden liian räikeitä ja häiritseviä oikean vaikutelman saamiseksi.

Varjon luomiseen on oma valaistuskomponentti, joka ei juurikaan valaise mallia, mutta luo havaittavan varjon mallin alaosaan. Tällä saatiin hieman pehmeämpi vaikutelma mallin valaistukselle.

Todellisen maailman valaistuksen huomioon ottamisen puuttuessa Vuforia-sovelluskehitysalusta, jouduttiin käyttämään näitä edellä mainittuja toimenpiteitä realistisemmän vaikutelman aikaan saamiseksi.



## Varjostukset

Mallin alle on luotu pinta pienen varjoreunuksen näyttämiseksi lattialla mallin vieressä. Pinnalle oli luotava oma materiaali, jolla saadaan itse pinta näkymättömäksi, mutta haluttu varjo näkymään sen päällä. Tämä tapahtui pintaan liitetyn materiaalin Shader-valikon UnlitShadows-valikosta löytyvän UnlitShadowReceive-osan päälle kytkemisellä, jonka jälkeen pohjan materiaali valikosta säädettiin Cutout-arvoksi 1.

## 5.7 Testaus

Sovellusta testattiin koko sen kehityskaaren ajan, sekä väliajoin sovelluksesta toimitettiin kehitysvaiheessa oleva versio myös toimeksiantajalle nähtäväksi ja testattavaksi. Sovelluksen kehityssuuntaa voitiin vielä näin hieman muokata sovelluksen toteutusvaiheessa, sekä lisätä vielä mahdollisesti ilmenneitä uusia toiminnallisuuksia sovellukseen. Joskin ennen työn aloitusta tehdyssä suunnitelma-dokumentissa oli pääpiirteet toteutettavasta sovelluksesta jo pitkälti selvillä.

Sovellusta testattiin Unity-pelimoottorin omassa editorissa, käyttäen valittavissa olevia eri kuvasuhteita. Kuin myös kahdella eri Android-käyttöjärjestelmän omaavalla mobiililaitteella: Sony Xperia Z2, Android-versio 6.0 ja Samsung Galaxy S8, Android-versio 7.0 ja 8.0. Näin saatiin näkökulmaa sovelluksen toimivuudesta myös vanhemmilla laitteilla, joilla mahdollisesti laskentateho on heikompaa ja esimerkiksi laitteen anturit voivat olla kehittymättömämpiä verrattuna nykyisin saatavilla olevaan tekniikkaan.

Testausta suoritettiin muun muassa seuraavilla osa-alueilla:

- mallin paikallaan pysyminen alustallaan
- sovelluksen sisäinen valaistus ja vaikutukset malliin
- reaali maailman valaistus ja sen vaikutukset merkkikuvan toimivuuteen
- mallin käyttäytyminen eri kuvakulmista tarkasteltuna
- valikkojen toimivuus
- asetusten toimivuus

- luukun aukeaminen ja sulkeutuminen halutulla tavalla
- tulen käyttäytyminen.

## 5.8 Sovelluksen käännös

Ennen sovelluksen käyttöä mobiililaitteissa, se täytyi kääntää halutulle alustalle sopivaksi. Android-käyttöliittymälle sovellusta kehitettäessä on viimeistään käännösvaiheessa varmistettava Android-SDK-sovelluskehittäjäpaketin asennuksen oltava tehty. Tosin Unity-pelimoottori ilmoittaa mahdollisista ohjelmistopuutteista tässä vaiheessa, mikäli käännöstä yritetään suorittaa puutteellisilla välineillä.

Käännettävän paketin nimi täytyi asettaa Player Settings-valikosta, sekä oli muistettava lisätä sovellukseen halutut kohtaukset mukaan ennen käännöstä. Tehdävä apk-käännöspaketti on syytä myös tallentaa tietokoneelle, mikäli käännöstä oltiin tekemässä suoraan kiinni kytkettyyn mobiililaitteeseen, muutoin käännöksen mennessä vain mobiililaitteen muistiin.

Sovelluksen käännettyä oikein ja apk-tiedoston valmistuttua, voitiin sovelluksen asennus tehdä liitteen 1 mukaan.

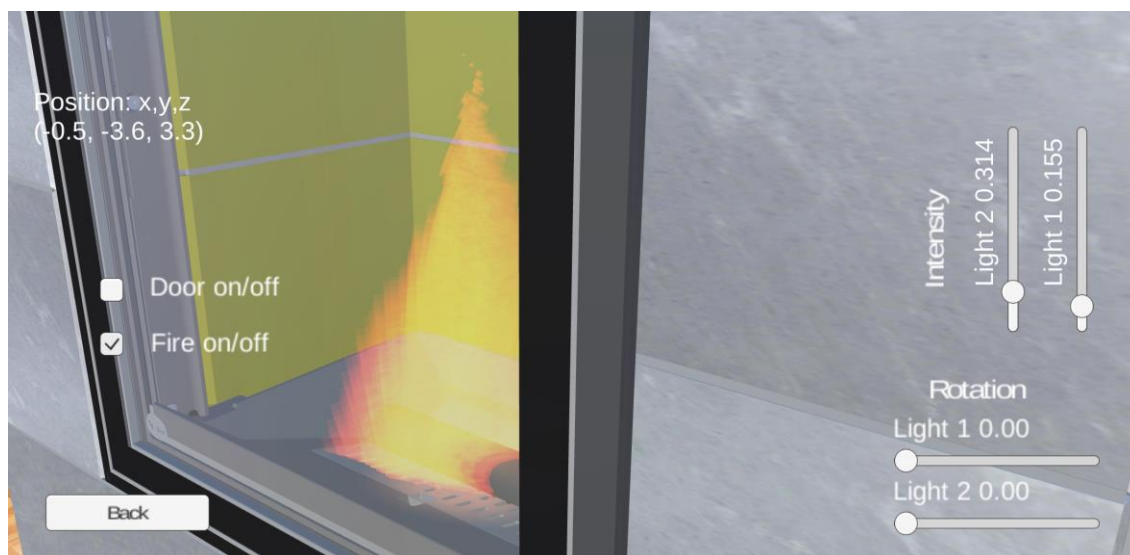
## 6 Tulokset

Lopputuloksena opinnäytetyön toiminnallisesta osuudesta saatiin sovellus, joka näyttää toimeksiantajan mallin luonnollisessa koossaan, lisätyn todellisuuden keinoin. Kuvassa (kuva 27) on esitetty lopullinen käyttöliittymänäkymä aloitusvalikosta, taustalla näkyvä malli pyörii hitaasti pystyakselinsa ympäri. Käyttöliittymä on visuaalisesti selkeä ja helppokäyttöinen, sen sisältäessä vain sovelluksesta poistumis-, info- ja käynnistyspainikkeet.



Kuva 27. Aloitusvalikko

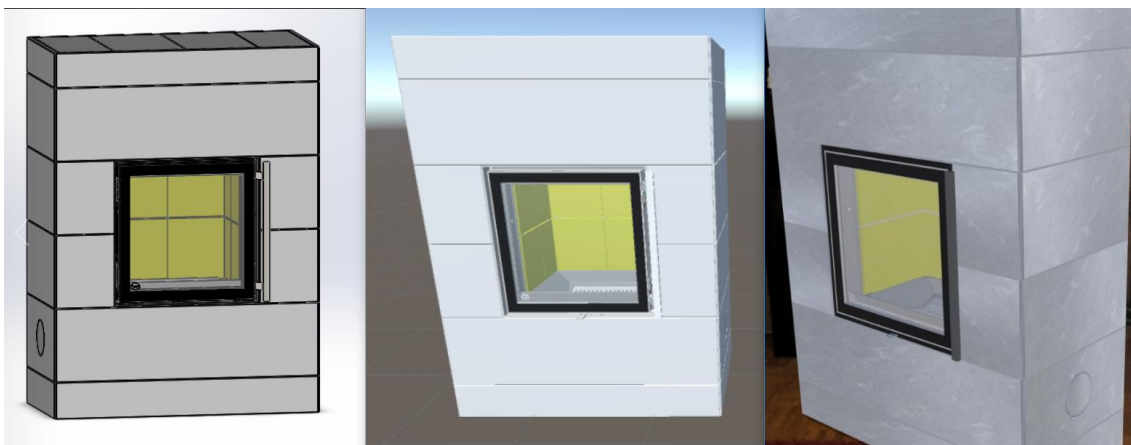
Myöskään itse sovelluksen käyttöliittymään ei tullut kuin asetukset ja poistumis-painikkeet, ja asetusvalikosta käyttäjä pääsee säätämään valaistusta, sekä luu-kun ja tulen toimintoja. Lisätyn todellisuuden sovellus toiminnassa (kuva 28), päällä näkyvissä asetusvalikon käyttöliittymä eri asetuksineen, varsinaisen sovel-luksen pyöriessä taustalla.



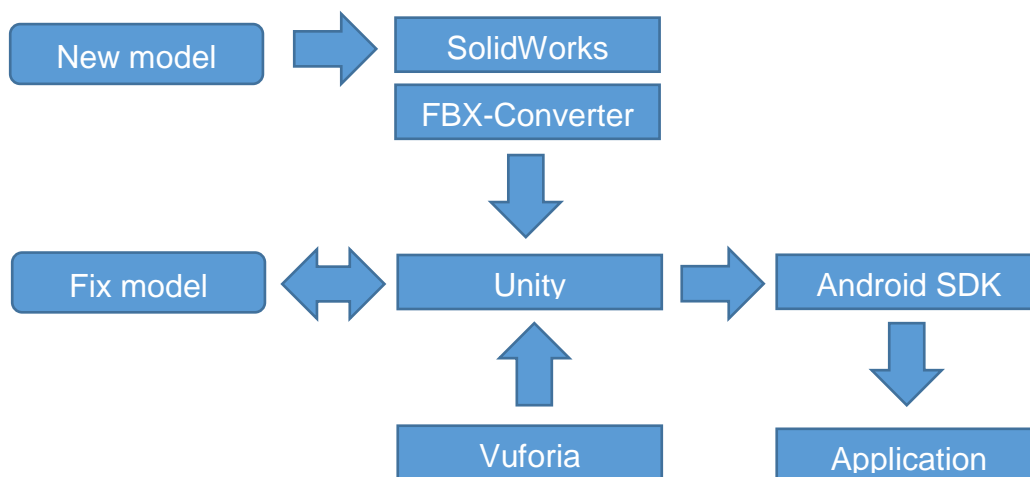
Kuva 28. Sovelluksen asetusvalikko

## 6.1 Prosessi

Mallin kehityskaari on nähtävissä kuvasarjasta (kuva 29), vasemmalla alkuperäinen SolidWorks malli, keskellä Unity-pelimoottoriin tuotu fbx-malli, ja oikealla malli, johon lisättyä pintamateriaalit ja valaistus.



Kuva 29. Mallin kehityskaari



Kuvio 1. Lisätyn todellisuuden sovelluksen prosessikaavio.

Uusi malli luodaan SolidWorks-ohjelmistolla, josta valmistunut tuotantokäyttöön suunnattu malli käännetään liitännäisellä Unity-sopivaan muotoon. Unity-pelimoottorissa mallia viimeistellään esittelysovelluksen käyttöön, sekä lisätään Vuforia tarjoamat lisätyn todellisuuden ominaisuudet. Lopputulos käännetään

valmiiksi sovellukseksi Unityllä, käyttäen pelimoottoriin liitettyä Android-sovelluskehityspakettia. (Kuvio 1.)

Vastaavien uusien mallien lisäys sovellukseen, sekä saattaminen sovelluksessa toiminta- ja esittelykuntoon tällä menetelmällä vie aikaa noin yhden työpäivän ajan. Mikäli malleja siirrettäisiin enemmänkin, aikavaatimus tulisi rutinoitumisen myötä pienentymään.

## 6.2 Yhteenveto

SolidWorksiin saatava liitännäinen, FBX-converter, on käytännöllisin ja yksinkertaisin vaihtoehto mallin siirtämiseksi Unity-pelimoottoriin. Siirrettävä malli pysyy sopivan kokoisena, säilyttäen kaikki siihen kuuluvat komponentit, lisäämättä mukaan kuitenkaan mitään ylimääräistä. Kyseinen liitännäinen on kuitenkin tehty juuri tähän haluttuun käyttötarkoitukseen ja lopputulos on aina parempi mitä pienemmillä välimuunnoksilla selvittää.

Mikäli joku kolmannen osapuolen mallinnusohjelma on jo valmiina käytössä, voidaan koettaa tehdä muunnosta myös sen avulla. Muunnosvaiheessa on huomioitava mahdolliset puutteet ja poistettava ylimääräiset lisäykset malliin. Mutta ulkopuolisen mallinnusohjelman käyttäminen on kuitenkin työläämpi tie, jos lopputuloksena halutaan saada pelkkä fbx-muunnos aikaan.

Mikäli lisätyn todellisuuden sovellus haluttaisiin saada ladattavaksi asiakkaille ja tuoda se markkinoille, tulee ottaa huomioon seuraavien sovellusten mahdolliset kustannukset ja käyttöoikeudet:

- Unity lisenssi
- Vuforia lisenssi
- SimLab FBX Exporter, kertamaksu

## 7 Pohdinta

Opinnäytetyön aikana kävi erittäin voimakkaasti ilmi lisätyn todellisuuden teknologioiden olevan voimakkaassa kehitys- ja nousuvaiheessa. Uusia teknologioita tuli yleisempään käyttöön työn tekemisen aikana, sekä jo olemassa olleet kehitysalustat saivat päivitettyjä versioita osakseen. Opinnäytetyö antoi myös uusia näkökulmia ja uutta tietoa lisätyn todellisuuden sovelluskehityksestä.

Etenkin sisustuksen suunnittelussa, kohteiden visualisoinneissa, sekä markkinoinnissa, pelien ja viihteen lisäksi, lisätty todellisuus ja sen sovellukset ovat todella käyttökelpoinen työkalu. Ilmi tuli myös sovellutuksia, joita voidaan käyttää hyödyksi esimerkiksi terveydenhoitoalalla ja oppimistarkoitukseen.

Kokonaisuutena opinnäytetyön tekeminen oli aiheen mielenkiintoisuuden ja ajan-kohtaisuuden vuoksi todella miellyttävää. Ennen työn aloitusta pohjatiedot aiheesta olivat jotakuinkin hyvällä tasolla, mutta myös uusia tekniikoita ja toimintatapoja tuli ilmi. Sain myös henkilökohtaisesti uusia ideoita ja ajatuksia lisätyn todellisuuden käyttömahdollisuuksista työn toteutuksen aikana. Useat käytettävät teknologiat tulivat paremmin tutuiksi ja aion käyttää, sekä seurata niiden kehitystä myös tulevaisuudessa.

### 7.1 Toteutuksen tarkastelu

Toimivaa lisätyn todellisuuden sovellusta voidaan hyödyntää toimeksiantajan myyntityössä ja mallistojen esittelykäytössä. Tulevaisuudessa myös toimeksiantajan muita mallistoja voitaisiin siirtää lisätyn todellisuuden sovelluksessa esiteltäväksi.

Toteutuneella työllä on positiivinen vaikutus toimeksiantajalle tuotteidensa markkinoinnissa. Myös Solidworks ja Unity-ohjelmistojen välisen yhteyden löytäminen

ja toimivaksi saattaminen on hyödyllinen myös tulevaisuudessa, mikäli mallia halutaan päästä hyötykäyttämään pelimoottorin avulla.

## **7.2 Tuotteen hyödyntäminen**

Toimeksiantajan tuotteet ovat kooltaan suuria ja painavia, joten niiden siirtäminen vaatii paljon työtä. Tuotteita ei näin ollen voida kuljettaa kätevästi mukana esimerkiksi messuilla tai myyntikäynneillä asiakkaiden luona. Lisätyn todellisuuden käyttäminen mallistojen esittelykäytössä on kustannustehokkaampaa kuin kokonaisen mallin siirtäminen ja rakentaminen esittelypaikalle.

Lisätyn todellisuuden sovelluksen avulla tuotteita voidaan esitellä nopeasti ja vertailla oikeassa koossaan paikasta ja ajasta riippumatta. Malleja voidaan tarkastella eri kulmista ja jopa kokeilla tuotteen toimintoja vuorovaikutteisella tavalla konkreettisesti lisätyn todellisuuden ympäristössä. Myös mallistojen esittelytiloissa voidaan esitellä aiempaa enemmän eri malleja ja niiden eri versioita, näiden viemättä kuitenkaan tilaa reaali maailmasta.

Sovellusta voidaan myös hyödyntää uusien mallien suunnittelussa. Mallista voidaan toteuttaa lisätyn todellisuuden esittelymalli jo ennen kuin ensimmäistäkään varsinaista konkreettista mallia on rakennettu tai valmistettu. Lisätyn todellisuuden mallin avulla saadaan tuotteesta parempi mielikuva kuin pelkältä tietokoneen ruudulta nähtynä. Sovellusta voitaisiin hyödyntää myös ennakkopalautteen saamisessa asiakkailta ja sidosryhmiltä jo suunnitteluvaiheen aikana.

## **7.3 Jatkotutkimus ja kehitysideat**

Näyttäisi että lisätyn todellisuuden sovelluskehitys ja siihen kytköksissä olevat kehitystyökalut ja lisäosat ovat tulossa suurempaan käyttöön

lähitulevaisuudessa. Myös alati kehittyvät mobiililaitteet helpottavat ja lisäävät suosiota lisätyn todellisuuden sovellusten kehittämisessä ja käytössä tulevaisuudessa.

Opinnäytetyön toteutuksen aikana pystyttiin jo testaamaan Googlen ARCore -kehitysalustaa, mutta sen ollessa vielä kehitysvaiheessa ja tekniikan tukiessa vasta muutamia eri mobiililaitteita, ei varsinaista sovellusta ollut tällä tekniikalla vielä järkevää toteuttaa. Joskin loppuvaiheessa työtä julkaistiin Google ARCore 1.0 ohjelmistoversio, joka toimi jo hieman useammassa laitteessa ja voidaankin olettaa tulevien versioiden lisäävän laitekantaa entisestään.

Myös Vuforia on tuomassa omaa ratkaisuaan merkittömän tunnistuksen parissa, tämä kehityshaara onkin ehdottomasti tulevaisuudessa seurattava asia. Vuforia Ground Plane -ominaisuuden tuleminen sovelluskehityskäyttöön juuri työn lopettamisen aikoihin on huomion arvoinen ominaisuus tulevaisuuden sovelluskehityksessä. Ominaisuuden tarjotessa samoja ominaisuuksia ARCore ja ARKit -kehityspakettien kanssa, ollessaan kuitenkin alustariippumaton.

Tällä hetkellä mallin lisäys sovellukseen vaati jonkin verran käsityötä, joten kehittämisen ja jatkotutkimuksen kohteena voisi olla myös mallin lataus sovellukseen ajon aikana. Ainakin seuraavia toimintoja voitaisiin tulevaisuudessa sovellukseen lisätä kohtuullisella vaivalla:

- Useita eri malleja sovelluksessa
- Mallin vaihto sovelluksen ajon aikana
- Mallin pintamateriaalien vaihto
- Mallin korkeuden muuttaminen
- Äänet
- Merkitön tunnistus

Myös virtuaalitodellisuuden sovellusta voisi testata tulevaisuudessa. Ainakin mobiililaitteille toteutettuna toimenpide olisi suhteellisen helppo toteuttaa Unity-peli-moottorista valmiina löytyvien ratkaisujen avulla. Eikä esimerkiksi sovelluksen siirtäminen HoloLens käyttöönkään vaatisi enää kohtuuttoman suuria ponnistuksia.



## Lähteet

1. Dassault Systèmes SolidWorks Corp. Tietoja SolidWorksista. 2018. [http://www.solidworks.fi/sw/6453\\_SVF\\_HTML.htm](http://www.solidworks.fi/sw/6453_SVF_HTML.htm). 21.3.2018.
2. Kielitoimiston sanakirja. Virtuaalitodellisuus. 2017. <https://www.kielitoimistonsanakirja.fi>. 3.4.2018.
3. Milgram, P. & Kishino. A. F. Taxonomy of Mixed Reality Visual Displays. 1994. [http://etclab.mie.utoronto.ca/people/paul\\_dir/IEICE94/ieice.html](http://etclab.mie.utoronto.ca/people/paul_dir/IEICE94/ieice.html). 3.4.2018.
4. Linowes, J. & Babilinski K. Augmented Reality for Developers. Birmingham: Packt Publishing Ltd. 2017.
5. State, J. VR, AR or MR...What's the Difference & Why Should I Care?. 2018. <http://www.appliedart.com/blog/vr-ar-or-mr-what-s-the-difference-why-should-i-care>. 7.3.2018.
6. Daugherty, P. Accenture. Extended Reality - Tech Vision 2018 Trend. 2018. <https://www.youtube.com/watch?v=wW0ncQl82ag>. 21.3.2018.
7. Baum, L. F. The Master Key. Indiana: Bowen-Merrill. 1901.
8. Mortonheilig.com. Sensorama patent. 2018. <http://www.mortonheilig.com/SensoramaPatent.pdf>. 17.3.2018.
9. Internet Archive. A head-mounted three-dimensional display\*. 2014. [https://web.archive.org/web/20140123204209/http://90.146.8.18/en/archiv\\_files/19902/E1990b\\_123.pdf](https://web.archive.org/web/20140123204209/http://90.146.8.18/en/archiv_files/19902/E1990b_123.pdf). 21.3.2018.
10. EL Habbak, H. & Cushnan, D. Developing AR Games for iOS and Android. Birmingham: Packt Publishing Ltd. 2013.
11. Matsuda, K. Hyper-Reality. 2016. <https://vimeo.com/166807261>. 21.3.2018.
12. Zuckerberg, M. Second Quarter 2016 Results Conference Call. 2016. [https://s21.q4cdn.com/399680738/files/doc\\_financials/2016/FB-Q216-Earnings-Transcript.pdf](https://s21.q4cdn.com/399680738/files/doc_financials/2016/FB-Q216-Earnings-Transcript.pdf). 21.3.2018.
13. Allen, B. Should you be using augmented reality in your marketing. 2017. <https://marketinginsidergroup.com/social-media/using-augmented-reality-marketing/>. 21.3.2018.
14. Shapiro, A. Huge. 6 Ways Marketers Can Get Augmented Reality Right. 2017. <http://www.adweek.com/digital/6-ways-marketers-can-get-augmented-reality-right/>. 21.3.2018.
15. Google Play. IKEA-Kuvasto. 2018. <https://play.google.com/store/apps/details?id=com.ikea.catalogue.android&hl=fi>. 2.3.2018.

16. Google Play. Porsche Mission E AR. 2018. <https://play.google.com/store/apps/details?id=com.porsche.missionear>. 21.3.2018.
17. Microsoft Corporation. Microsoft HoloLens. 2018. <https://www.microsoft.com/fi-fi/hololens>. 21.3.2018.
18. Oculus VR. Oculus Rift. 2018. <https://www.oculus.com/rift/>. 21.3.2018.
19. PTC. Getting Started. 2018. <https://library.vuforia.com/>. 21.3.2018.
20. Hololamp. Hololamp. 2018. <http://hololamp.tech/>. 21.3.2018.
21. Tulikivi Oyj. Tulikivi Vuosikertomus 2017. 2018. [http://www.tulikivi.fi/julkaisut/yritys/vuosiker/fi/ES2020\\_vk2017\\_fin.pdf](http://www.tulikivi.fi/julkaisut/yritys/vuosiker/fi/ES2020_vk2017_fin.pdf). 24.3.2018.
22. Prättälä, M. 2017. Myyntijohtaja. Tulikivi Oyj. Aloitustapaaminen, Heinäveden tehdas 2.11.2017.
23. Moilanen, M. 2017. Design Manager & Mäkkeli, J. 2017. Pääsuunnittelija. Tulikivi Oyj. Suunnittelijoiden tapaaminen, Juuan tehdas 10.11.2017.
24. Tulikivi Oyj. Raita. 2018. <http://www.tulikivi.fi/tuotteet/RAITA>. 22.3.2018.
25. Fine, R. UnityScript's long ride off into the sunset. 2017. <https://blogs.unity3d.com/2017/08/11/unityscripts-long-ride-off-into-the-sunset/>. 22.3.2018.
26. Unity. Unity, multiplatform. 2018. <https://unity3d.com/unity/features/multiplatform>. 22.3.2018.
27. Unity. Unity Store. 2018. <https://store.unity.com/>. 22.3.2018.
28. Unity. Materials, Shaders & Textures. 2018. <https://docs.unity3d.com/Manual/Shaders.html>. 22.3.2018.
29. Thorn, A. Mastering Unity 2017 Game Development with C#: Create professional games with solid gameplay features and professional-grade workflow. Birmingham: Packt Publishing Ltd. 2017.
30. Google. ARCore Overview. 2018. <https://developers.google.com/ar/discover/>. 22.3.2018.
31. Kastrenakes, J. Google's Project Tango is shutting down because ARCore is already here. 2017. <https://www.theverge.com/2017/12/15/16782556/project-tango-google-shutting-down-arcore-augmented-reality>. 22.3.2018.
32. projecttango. 2017. "We're turning down support for Tango on March 1, 2018. Thank you to our incredible community of developers who made such progress with Tango over the last three years. We look forward to continuing the journey with you on ARCore." Twiitti. 22.3.2018.
33. Apple. ARKit. 2018. <https://developer.apple.com/arkit/>. 22.3.2018.

34. PTC. Image Targets. 2018. <https://library.vuforia.com/articles/Training/Image-Target-Guide>. 22.3.2018.
35. PTC. Introduction to Ground Plane in Unity. 2018. <https://library.vuforia.com/articles/Solution/ground-plane-guide.html>. 22.3.2018.
36. PTC. Pricing. 2018. <https://developer.vuforia.com/vui/pricing>. 22.3.2018.
37. SimLab Soft. SolidWorks plugin: FBX Exporter. 2018. [http://www.simlab-soft.com/3d-plugins/SolidWorks/fbx\\_exporter\\_for\\_SolidWorks-main.aspx](http://www.simlab-soft.com/3d-plugins/SolidWorks/fbx_exporter_for_SolidWorks-main.aspx). 22.3.2018.
38. PTC. Vuforia Developer, downloads. 2018. <https://developer.vuforia.com/downloads/sdk>. 25.3.2018.
39. Unity. Designing UI for Multiple Resolutions. 2018. <https://docs.unity3d.com/Manual/HOWTO-UIMultiResolution.html>. 25.3.2018.
40. Tulikivi Oyj. Logot ja muut kuvat. 2018. [http://www.tulikivi.fi/tulikivi/logot\\_ja\\_muut\\_kuvat](http://www.tulikivi.fi/tulikivi/logot_ja_muut_kuvat). 26.3.2018.

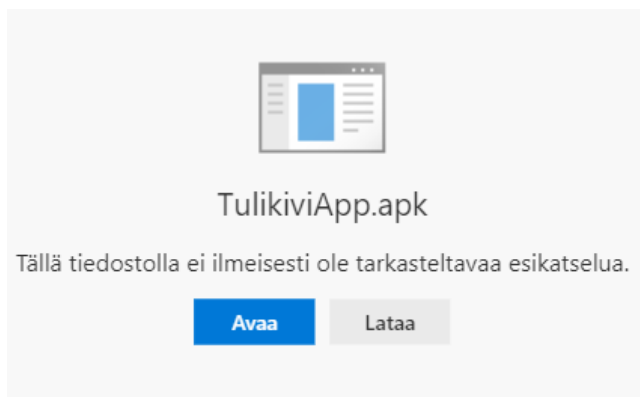
## ASENNUSOHJE

Tulikivi AR-sovelluksen asennusohje Android laitteille.

Asennettuna sovellus vie tilaa 75.59 Mt.

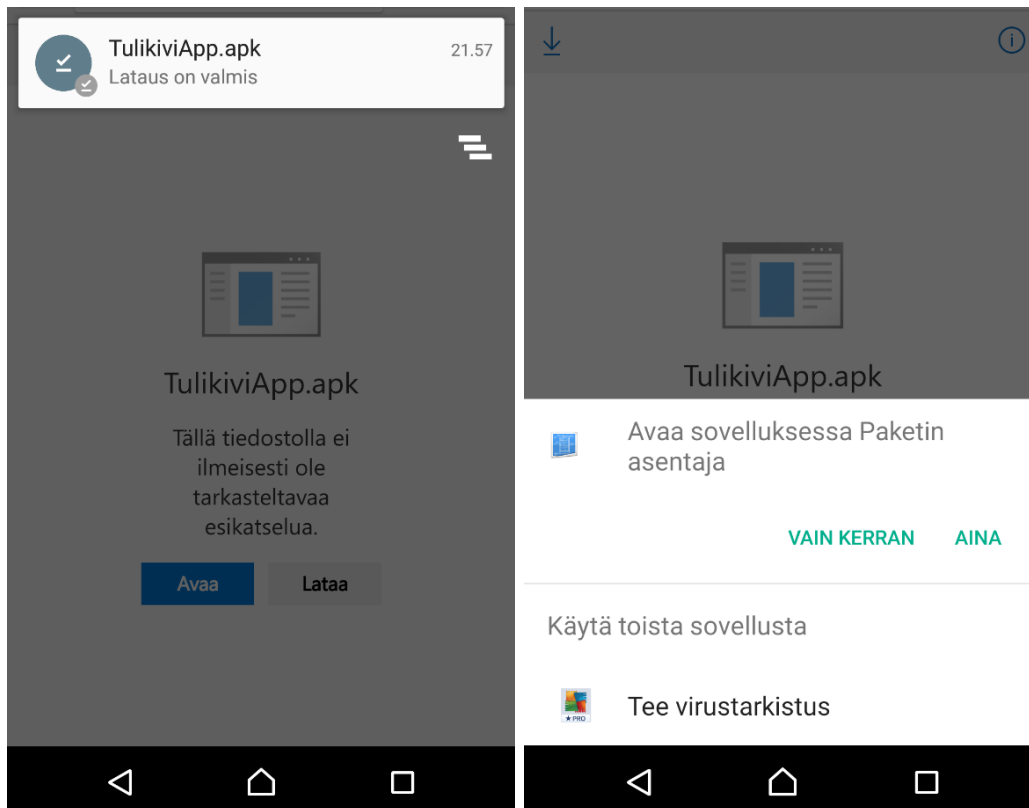
Mene laitteella johon haluat asentaa sovelluksen alla olevaan osoitteeseen ja lataa sovellus.

<https://urly.fi/W4L>

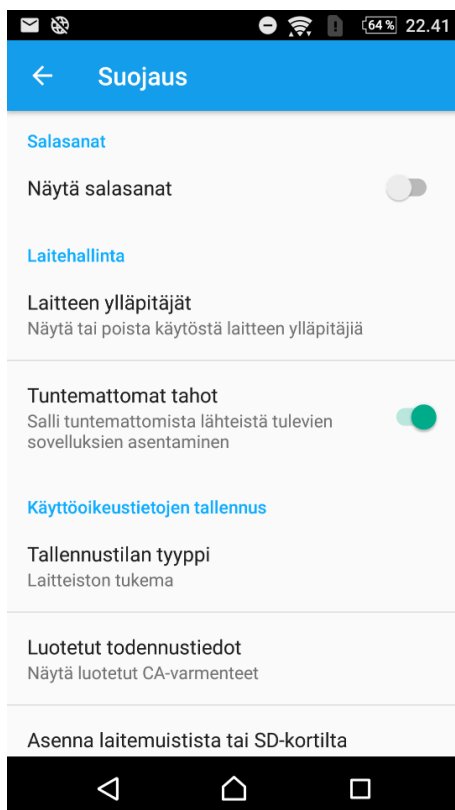


Valitse Lataa (Avaa johtaa samaan lopputulokseen). (Paketin nimi on TulikiviAR.apk)

*"Mikäli lataat tiedoston tietokoneelle, voit myös siirtää sen käsin Android laitteeseen. Yhdistä laite tietokoneeseen ja siirrä tiedosto haluamaasi paikkaan mobiililaitteessa josta löydät sen, tai siirrä tiedosto muistikortille. Riippuen laitteesta, tiedostojen ja hakemistojen selailu laitteessa voi olla hieman hankalaa. Siirron jälkeen voit seurata asennusohjeita normaalisti."*

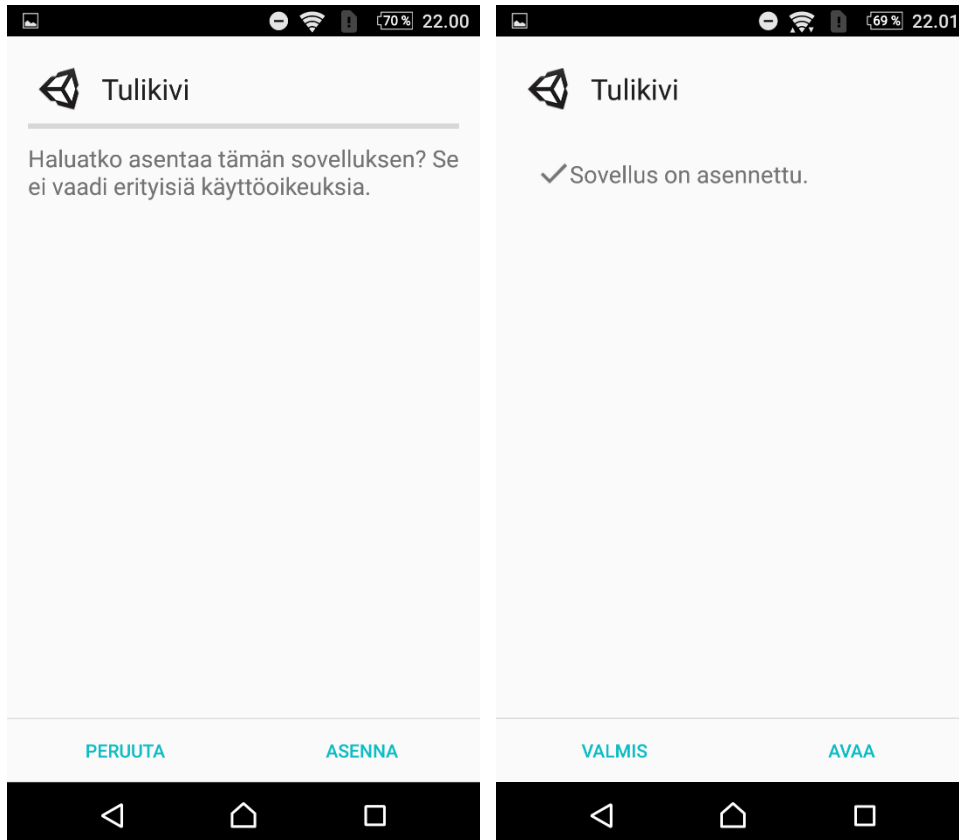


Kun lataus on valmis, klikkaa ladattua kohdetta ja aloita sovelluksen asennus.



Laitteen asetuksista pitää sallia sovelluksen asennus tuntemattomista lähteistä. Asetukset -> Suojaus -> Tuntemattomat tahot.

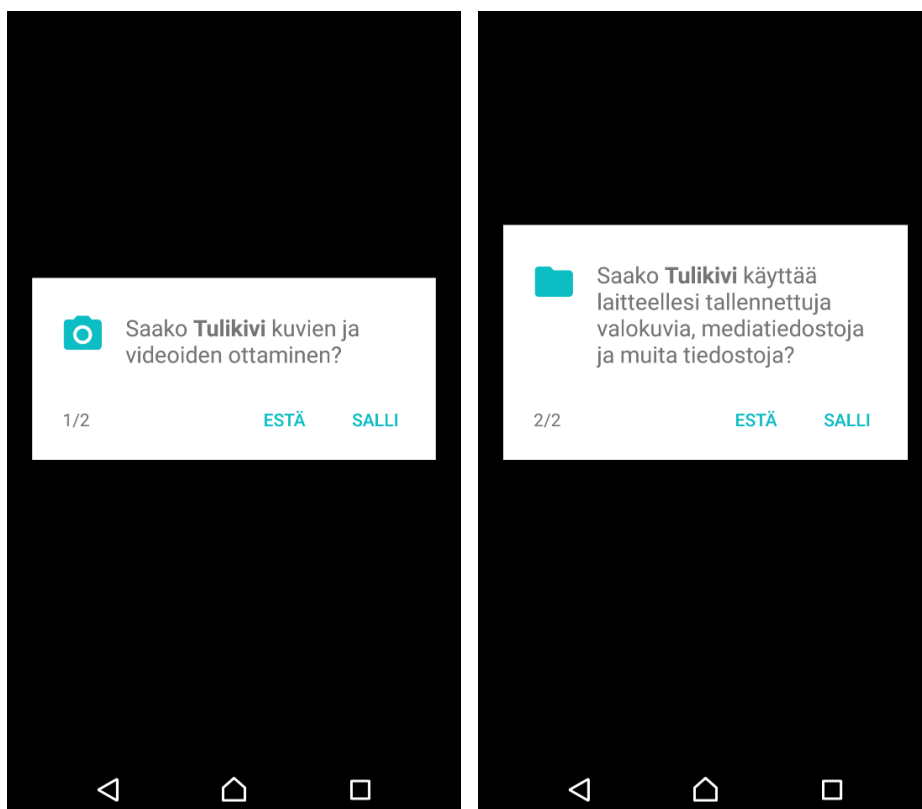
Vanhemman mallisilla laitteilla tämä asetus pitää käydä itse muuttamassa, uudemmat laitteet vievät käyttäjän suoraan asetuksen luo. (*"Eri laitteiden välillä valikot ja asetukset voivat hieman vaihdella, mutta pääpiirteittäin periaate on tämä"*)



Valitse "Asenna". Kun sovellus on asennettu, valitse "Avaa".



Mikäli et halua vielä avata sovellusta, se löytyy myös asennuksen jälkeen laitteen työpöydältä tai valikoista.



Ensimmäisen käynnistyksen yhteydessä, salli sovellukselle käyttöoikeudet.



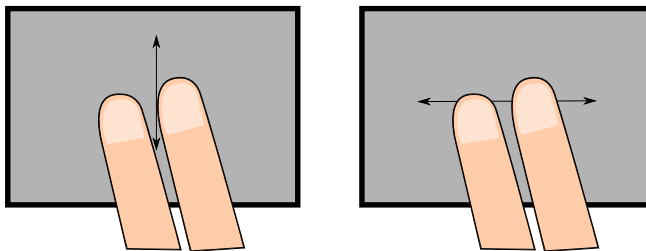
Kun sovellus on käynnistynyt, paina Start-painiketta käynnistääksesi AR-sovelluksen.

Tulosta liite 5 Tulikivi\_tunnus\_BLACK.JPG [40] kuva (A4, mustavalko) ja osoita sovel-  
luksen kameranäkymä kuvaa kohti.



### Mallin liikuttaminen

Kun malli on ilmestynyt kuvan päälle, voit halutessasi kääntää mallia liu'uttamalla  
kahta sormeä näytöllä.



### Poistuminen sovelluksesta

Voit poistua AR-osiosta painamalla Menu kuvaketta.



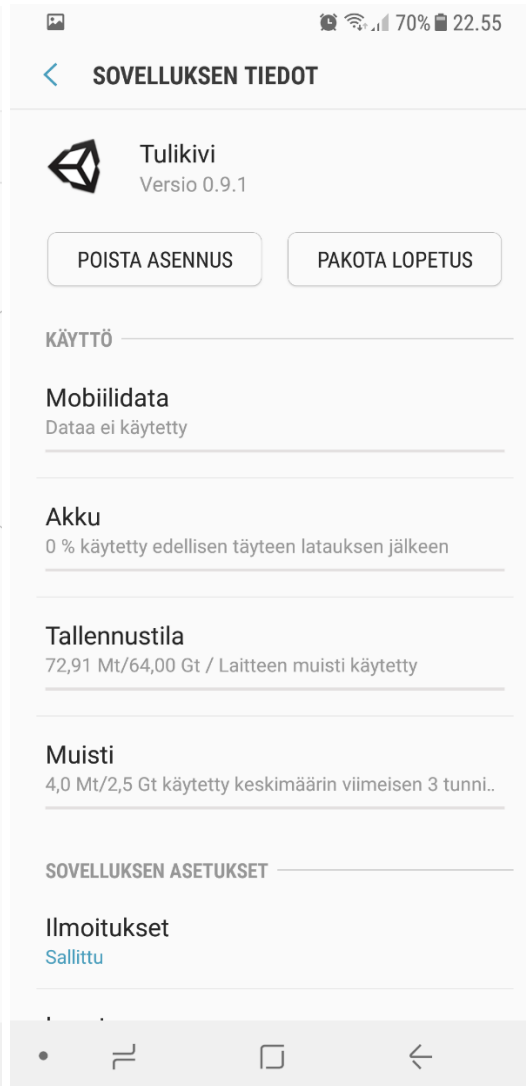
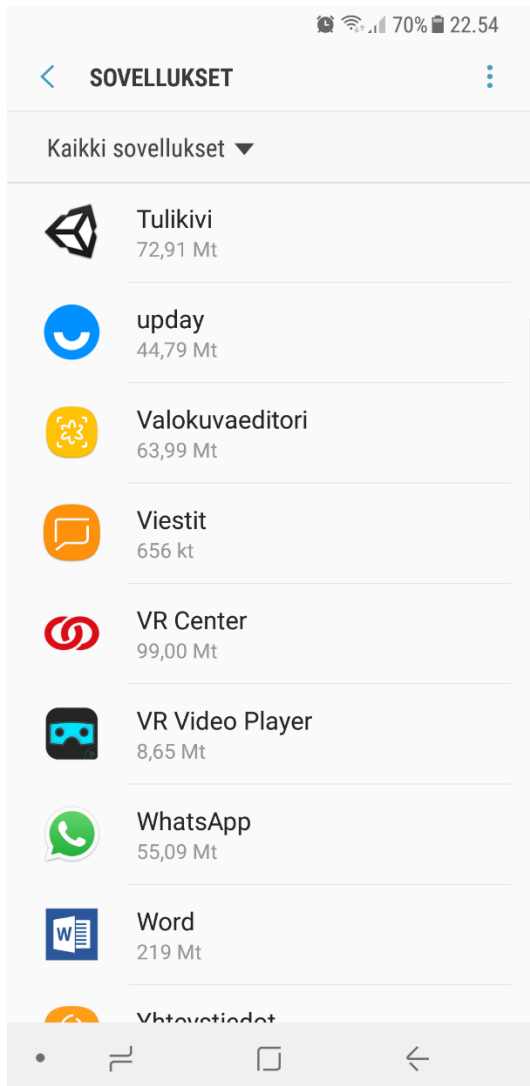
Ja koko sovelluksesta painamalla Quit kuvaketta.



### Sovelluksen poistaminen

Sovelluksen voit poistaa laitteesta valitsemalla laitteesta Asetukset -> Sovellukset ->  
Tulikivi -> Poista Asennus. (*"Eri laitteiden välillä valikot ja asetukset voivat hieman  
vaihdella, mutta pääpiirteittäin periaate on tämä"*)





```

//OpenDoor.cs
using UnityEngine;

public class OpenDoor : MonoBehaviour {

    #region ROTATE
    private float _sensitivity = 0.4f;
    private Vector3 _mouseReference;
    private Vector3 _mouseOffset;
    private Vector3 _rotation = Vector3.zero;
    private bool _isRotating;
    public GameObject Target;
    private float angle;
    public bool LockDoor;
    #endregion

    public void ResetDoor ()
    {
        Target.transform.localEulerAngles = new Vector3(0, 90, 0); //pa-
        lautetaan luukku kiinni
    }

    void Update() //close point 90, open point max 195
    {
        if (LockDoor) {
            angle = Target.transform.localEulerAngles.y;

            if (_isRotating)
            {
                // offset
                _mouseOffset = (Input.mousePosition - _mouseReference);

                // apply rotation
                _rotation.y = -(_mouseOffset.x + _mouseOffset.y) * _sensi-
            tivity;

            if (angle < 90) //pysäytetään luukun liikkuminen kiinni
            mennessä
            {
                Target.transform.localEulerAngles = new Vector3(0, 90,
                0); //palautetaan luukku y-kulman 90-asteeseen

                //angle = 90;

                //_isRotating = false;
            }
            else if (angle > 195) //pysäytetään luukun liikkuminen
            auki mennessä
            {
                Target.transform.localEulerAngles = new Vector3(0, 195,
                0); //palautetaan luukku y-kulman 195-asteeseen

                //angle = 195;

                //_isRotating = false;
            }
            if (angle >= 90 && angle <= 195)
            {
                // rotate
                Target.transform.Rotate(_rotation);
            }
        }
    }
}

```

```

        // store new mouse position
        _mouseReference = Input.mousePosition;
    }
    else
    {
        _isRotating = false;
    }
}

if (Input.GetMouseButtonDown(0))
//void OnMouseDown()
{
    // rotating flag
    _isRotating = true;
    // store mouse position
    _mouseReference = Input.mousePosition;
}

//void OnMouseUp()
if (Input.GetMouseButtonUp(0))
{
    //angle = 91;
    // rotating flag
    _isRotating = false;
}
}
}
}
}

```

```

//MenuScript.cs
using UnityEngine;
using UnityEngine.SceneManagement;
using Vuforia;

public class MenuScript : MonoBehaviour {

    public Canvas MenuCanvas;
    public Canvas InfoCanvas;

    void Start()
    {
        VuforiaRuntime.Instance.Deinit(); //need this to close ARCamera
        when coming back to menu
    }

    void Awake()
    {
        InfoCanvas.enabled = false;
    }

    public void InfoBtn()
    {
        InfoCanvas.enabled = true;
        MenuCanvas.enabled = false;
    }

    public void BackBtn()
    {
        InfoCanvas.enabled = false;
        MenuCanvas.enabled = true;
    }

    public void StartBtn()
    {
        //Application.LoadLevel(1);

        VuforiaRuntime.Instance.InitVuforia(); //Vuforia ARCamera starts

        SceneManager.LoadScene(1);
    }
}

```

```

//SmoothCamera.cs
using UnityEngine;
using System.Collections.Generic;
using Vuforia;

[RequireComponent(typeof (VuforiaBehaviour))]
public class SmoothCamera : MonoBehaviour {

    public int smoothingFrames = 10;
    private VuforiaBehaviour qcarBehavior;

    private Quaternion smoothedRotation;
    private Vector3 smoothedPosition;

    private Queue<Quaternion> rotations;
    private Queue<Vector3> positions;

    public void OnInitialized() {
    }

    public void OnTrackablesUpdated() {

        if (rotations.Count >= smoothingFrames) {
            rotations.Dequeue();
            positions.Dequeue();
        }

        rotations.Enqueue(transform.rotation);
        positions.Enqueue(transform.position);

        Vector4 avgr = Vector4.zero;
        foreach (Quaternion singleRotation in rotations) {
            Math3d.AverageQuaternion(ref avgr, singleRotation, rotations.Count);
        }

        Vector3 avgp = Vector3.zero;
        foreach (Vector3 singlePosition in positions) {
            avgp += singlePosition;
        }
        avgp /= positions.Count;

        smoothedRotation = new Quaternion(avgr.x, avgr.y, avgr.z, avgr.w);
        smoothedPosition = avgp;
    }

    // Use this for initialization
    void Start () {

        rotations = new Queue<Quaternion>(smoothingFrames);
        positions = new Queue<Vector3>(smoothingFrames);
        qcarBehavior = GetComponent<VuforiaBehaviour>();

        VuforiaARController vuforia = VuforiaARController.Instance;
        // qcarBehavior.RegisterVuforiaStartedCallback(OnInitialized);
        // qcarBehavior.RegisterTrackablesUpdatedCallback(OnTrackablesUpdated);

        vuforia.RegisterVuforiaStartedCallback(OnInitialized);
        vuforia.RegisterTrackablesUpdatedCallback (OnTrackablesUpdated);
    }
}

```

```
// Update is called once per frame
void LateUpdate () {
    transform.rotation = smoothedRotation;
    transform.position = smoothedPosition;
}
}
```

